平成19年度経済産業省委託事業成果

平成19年度基準認証研究開発委託事業

バイオメトリクスアプリケーションインターフェース標準規格へのアプリケーションの適合性試験に関する標準化

平成20年３月

社団法人日本自動認識システム協会

まえがき

平成１９年度、基準認証研究開発事業として、経済産業省からの委託事業「バイオメトリクスアプリケーションインターフェース標準規格へのアプリケーションの適合性試験に関する標準化」の成果報告書である。本テーマは平成19～２１年度まで、3年間の予定で計画されているもので初年度に実施し、社団法人日本自動認識システム協会において実施されたものである。

＜目　　次＞

1　総論……6

2　委託研究実施計画……6

2．1　事業の目的……6
2．2　事業内容……7
2．2．1　国際規格化の活動……7
2．2．2　ＩＳＯ／ＩＥＣ　ＳＣ３７国際会議および成果の公表……7
2．2．3　ＢｉｏＡＰＩを初めとするＳＣ３７の適合性評価の実運用方法の調査……7
2．3　実施場所……7
2．4　実施日程……8
2．5　委員会開催……9
2．6　研究体制……10
2．7　成果の概要……11
2．7．1　適合性試験プログラムの開発（ＢｉｏＡＰＩフレームワークのテストプログラム）……12
2．7．2　ＩＳＯ／ＩＥＣ２４７０９関連国際規格の解析……12
2．7．3　ＩＳＯ／ＩＥＣ　ＳＣ３７国際会議および成果の公表……13
2．7．4　　ＢｉｏＡＰＩを初めとするＳＣ３７の適合性評価の実運用方法の調査……13
2．7．5　まとめ……14

3　関連国際規格の調査及び解析……16

3．1　ISO／IEC19784－1:2006Information technology－Biometric application programming interface－Part 1:BioAPI specificationについて……16
3．1．1　はじめに……16
3．1．2　ＢｉｏＡＰＩの構造……16
3．1．3　ＢｉｏＡＰＩのデータ型とマクロについて……27
3．1．4　ＢｉｏＡＰＩ関数仕様……27
3．1．5　ＢｉｏＡＰＩバイオメトリックサービスプロバイダー（ＢＳＰ）について……27
3．1．6　適合性評価について……28
3．2　BioAPI Liteについて……44
3．2．1　はじめに……44
3．2．2　New Work Item Proposal(NWI)の内容……44
3．2．3　ベースドキュメントについて……44
3．3　ISO／IEC24709－1:2007Information technolog－Conformance testing for the biometric application programming interface(BioAPI)－Part1:Methods and proceduresについて……52

３．３．１　はじめに.....52
３．３．２　24709-1の文書構成.....52
３．３．３　24709-1で説明している内容.....53
３．３．４　適合性評価試験方法.....53
３．３．５　主張言語.....54
３．３．６　標準インターフェース.....54
３．３．７　適合性評価試験の具体的な進め方.....56
３．４　ISO／IEC24709－2:2007 INFORMATION TECHNOLOGY－CONFORMANCE TESTING FOR BIOAPI－PART2:TEST ASSERTION FOR BIOMETRIC SERVICE PROVIDERSについて.....56
３．４．１　はじめに.....56
３．４．２　適用範囲.....56
３．４．３　適合性について.....57
３．４．４　引用文献.....57
３．４．５　一般原則.....57
３．４．６　ＢｉｏＡＰＩＢＳＰの適合性評価試験（テストアサーション）.....57
３．４．７　適合性評価試験詳細仕様.....78
４　ＢＩＯＡＰＩ２．０フレームワークの適合評価試験仕様.....78
４．１　はじめに.....78
４．２　適用範囲.....78
４．３　用語の意味.....78
４．４　基本仕様.....80
４．４．１　適合試験の方法論.....81
４．４．２　適合試験の方法.....93
４．４．３　適合性の判定方法.....95
４．５　詳細仕様.....95
４．５．１　はじめに.....95
４．５．２　適用範囲.....95
４．５．３　試験詳細（テストアサーション）.....95
５　OSS NOKALVA BIOFOUNDRY社製 BIOAPI2.0準拠フレームワークを活用したテスト.....96
５．１　はじめに.....96
５．２　ソフトウェア構成.....97
５．３　動作テスト.....99
５．３．１　ＯＳＳ推奨の動作テスト.....100
５．３．２　セキュアデザイン研究所の指紋認証装置を利用した動作テスト.....105
(FP-PLUS指紋収集装置の説明).....105
５．４　テスト結果.....132

５．５　今後の予定 132
６　ＢＩＯＡＰＩをはじめとするＳＣ３７の適合性評価の実運用方法の調査 133
６．１　調査研究の目的 133
６．１．１　目的 133
６．２　適合性評価制度の概要 135
６．２．１　適合性評価制度について 135
６．２．２　適合性評価統一の国際動向 136
６．２．３　国際標準と適合性評価委員会CASCO 141
６．２．４　適合性評価体制の具体例 142
６．２．５　まとめ 144
６．３　各国の適合性評価制度調査 148
６．３．１　韓国 148
６．３．２　シンガポール 163
６．３．３　ＥＵ 168
６．３．４　米国 175
６．３．５　日本 192
６．３．６　まとめ 195
６．４　バイオメトリクスに関する適合性評価対象 198
６．４．１　開発案件 198
６．５　日本における適合性評価体制 201
６．５．１　適合性評価における標準化の状況 201
６．５．２　社会的な背景 202
６．５．３　日本のとるべき対応 206
６．６　まとめ 209

平成１９年度経済産業省　産業技術研究開発委託事業

「バイオメトリクスアプリケーションインターフェース標準規格への
アプリケーションの適合性試験に関する標準化」

1　総論

近年、外国人テロリストの侵入防止、本人成りすまし等、偽造手段への強力な対策技術として、世界各国においてバイオメトリクス（生体認証技術）を社会インフラとして急速に導入する動きが展開してきている。パスポート、自動車運転免許証への応用が開始されており、国内においても、世界各国との連携を考慮したこれらの導入をはじめ、金融機関のキャッシュカードやＡＴＭへの利用が急速に進みつつある。ＩＳＯ／ＩＥＣはこのようなバイオメトリクス進展の流れをとらえて、２００２年より、バイオメトリクスの基幹技術の国際標準を策定するための活動を開始し､昨年､バイオメトリックスを応用したシステムの構築に必須となるアプリケーションインターフェース（以下ＢｉｏＡＰＩ）の国際標準(ＩＳＯ／ＩＥＣ１９７８４ＢｉｏＡＰＩ)を策定した。

さらに、ＩＳＯ／IＥＣは、この標準規格への適合性の検証方法を国際標準として２００６年公布した。（ＩＳＯ／ＩＥＣ　２４７０９－１，２）

ＩＳＯ／IＥＣは国際標準規格の策定と共に、そのバイオメトリクスを応用した製品が、その規格に適合しているかどうかの適合試験、およびその認証機関の設置を各国が推進することを推奨している。

本研究テーマではＢｉｏＡＰＩの規格に定義されているフレームワークおよびアプリケーションに準拠した製品がＩＳＯ／ＩＥＣ１９７８４－１：２００６に適合しているかどうかの評価を実施する方法を具体的に決めるための実証モデルを作成し，これをもとに国際規格ＩＳＯ／ＩＥＣ　２４７０９－３，４の策定に貢献することを目標とする。

また，ＩＳＯ／ＩＥＣ　ＪＴＣ１　ＳＣ３７においての適合試験はＩＳＯ／ＩＥＣ　ＪＴＣ１　ＳＣ３７　ＷＧ２　ＢｉｏＡＰＩ関連のみならずＩＳＯ／ＩＥＣ　ＪＴＣ１　ＳＣ３７　全体においても議論が必要であるため，今後の日本のスタンスについても検討する。

2　委託研究実施計画

２．１　事業の目的

バイオメトリックシステムが普及展開していく中で，さまざまなバイオメトリクス製品が開発されていくが、その一方で製品が国際規格に準拠している必要がある。　そのため、本事業ではこれらの製品が国際規格ＢｉｏＡＰＩに準拠しているかどうかを適合評価するための第三者による適合評価試験を実施する時代に向けたテーマを選択した。

特に、国内の製品のみならず、海外製品の導入時、および海外市場への国内ベンダーによる輸出時に国際規格への適合保証が出来る国内インフラ構築は欠かせないものである。このため３年

計画の中で本年度は下記項目を目的として研究テーマを実施する。

（1）国際規格ＩＳＯ／ＩＥＣ１９７８４（ＢｉｏＡＰＩ　２．０）に準拠したバイオメトリクスのシステムがこの国際規格に適合していることを検証する試験方法を構築し、国際規格として提案する。（日本としてＩＳＯ/IＥＣへの貢献）

（2）　ＩＳＯが推奨する方式（ＩＳＯ　ＣＡＳＣＯ）に則り、国内に於ける本規格（ＩＳＯ／ＩＥＣ１９７８４）への適合性を検証し、公正な認定、公開の方法、組織等、実運用に向けた調査を開始する。

２．２　事業内容

２．２．１　国際規格化の活動

（1）ＩＳＯ／ＩＥＣ２４７０９－３策定作業

（2）適合性試験プログラムの開発（ＢｉｏＡＰＩフレームワークのテストプログラム）

・レファレンスフレームワーク入手・解析動作確認

・プログラム仕様検討

・プログラム基本設計

・プログラム詳細設計

（3）ＩＳＯ／ＩＥＣ２４７０９関連国際規格の解析

・ＩＳＯ／ＩＥＣ１９７８４－１（ＢｉｏＡＰＩ）

・ＩＳＯ／ＩＥＣ２４７０９－１，２

（4）ＩＳＯ／ＩＥＣ２４７０９－４の策定作業および適合性試験プログラムの開発検討

２．２．２　ＩＳＯ／ＩＥＣ　ＳＣ３７国際会議および成果の公表

ＢＣ（バイオメトリクスカンファレンス）およびＡＢＣ(アジア・バイオメトリクス・コンソーシアム)での情報収集、Ｈ２０年１月のＳＣ３７　イスラエル会議等での公表

２．２．３　ＢｉｏＡＰＩを初めとするＳＣ３７の適合性評価の実運用方法の調査

・ＩＳＯ適合性試験の調査(ＣＡＳＣＯ、他のＩＳ試験認定機関のあり方)

２．３　実施場所

実施場所：社団法人日本自動認識システム協会

東京都千代田区岩本町１－９－５　ＦＫビル７階

分室　　社団法人日本自動認識システム協会セキュアデザイン研究所分室

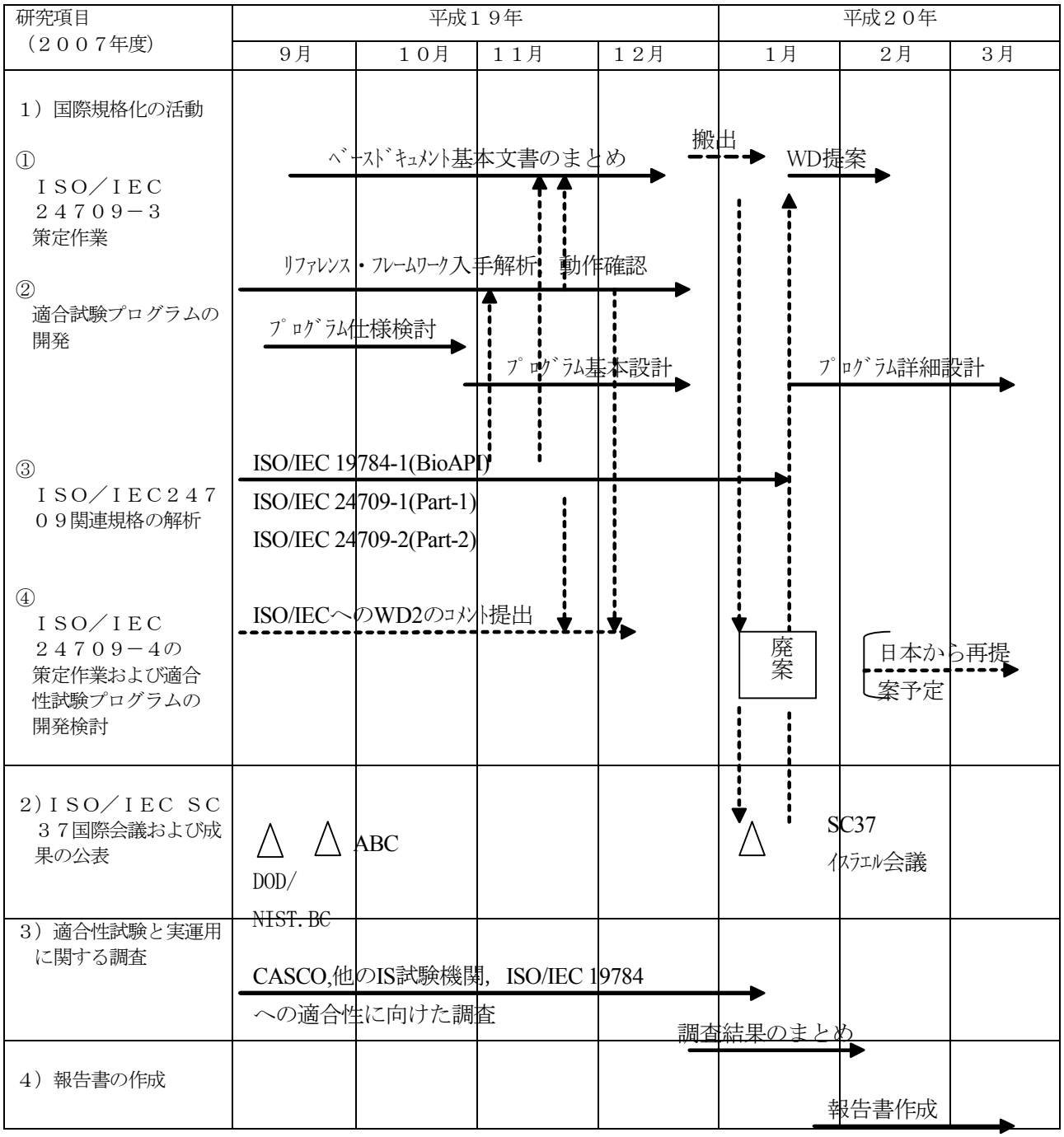
東京都港区芝２－８－１３　キタハイム芝ビル３Ｆ

分室　　社団法人日本自動認識システム協会日立情報制御ソリューションズ分室

茨城県日立市大みか町５丁目２番１号

２．４　実施日程

実施期間：受託契約締結日から平成２０年３月３１日まで



２．５　委員会開催

「バイオメトリクスアプリケーションインターフェース標準規格へのアプリケーションの適合性試験に関する標準化」の審議委員会を社団法人日本自動認識システム協会にて実施。委員会開催スケジュールを示す。

委員会名	８月	９月	１０月	１１月	１２月	１月	２月	３月
１．審議委員会			方針 確認		中間 確認 ▽		最終確認	▽

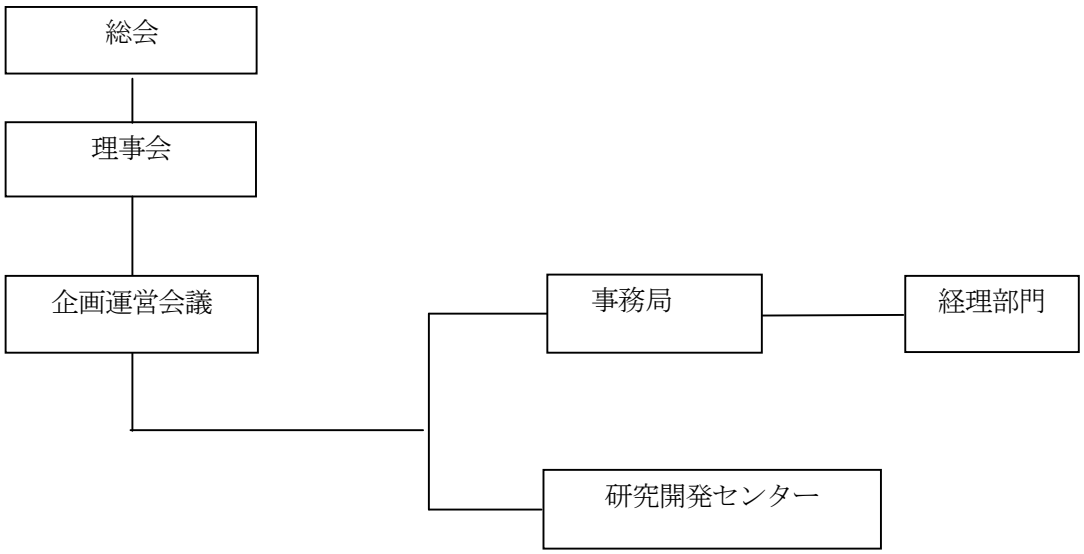
（委員会構成メンバー）

役割	氏名	所属	役職
委員長	半谷精一郎	東京理科大学工学部電気工学科	教授
委員	和泉　章	経産省産業技術環境局情報電気標準化推進室	室長
委員	鈴木　晴光	経済産業省製造産業局産業機械課	課長補佐
委員	山田　安秀	独立行政法人情報処理推進機構	センタ長
委員	大塚　玲	独立行政法人産業総合研究所	情報セキュリティ研究センタ　チーム長
委員	秋間　升	(財)日本規格協会　INSTAC	所長
委員	木元　和宏	(財)日本規格協会(JSA)	調査役
委員	廣川　勝久	(株)電子商取引安全技術研究所	SC17委員長
委員	瀬戸　洋一	SC37国内委員会	SC37委員長
委員	国分　明男	ニューメディア開発協会	首席研究員
委員	鷲見　和彦	三菱電機　先端技術総合研究所	SC37WG5主査
委員	中村　敏男	沖電気株式会社　情報通信事業G	SC37WG２主査
委員	山田　朝彦	東芝ソリューション（株）	SC27&SC37 WG2委員
委員	新崎　卓	富士通研究所　画像バイオメトリクス研究センタ	SC37WG3主査
委員	坂本　静生	日本電気株式会社　メディア情報研究所	SC17主査
委員	三村　昌弘	日立製作所　情報通信グループ	SC27委員
委員	池野　修一	セコム　IS研究所	SC37WG6主査
オブザーバ	森田　信輝	経産省産業技術環境局情報電気標準化推進室	室長補佐
オブザーバ	内藤　貴浩	経済産業省製造産業局産業機械課	係長
推進関係者	栗田　寛久	セキュアデザイン研究所	JAISA研究員
推進関係者	宇都宮康夫	日立情報制御ソリューションズ	JAISA研究員
事務局	大木　勝雄	社団法人日本自動認識システム協会	専務理事
事務局	武田　忠夫	社団法人日本自動認識システム協会	特別顧問
事務局	中嶋　晴久	社団法人日本自動認識システム協会	
事務局	柴田　夏彦	社団法人日本自動認識システム協会	

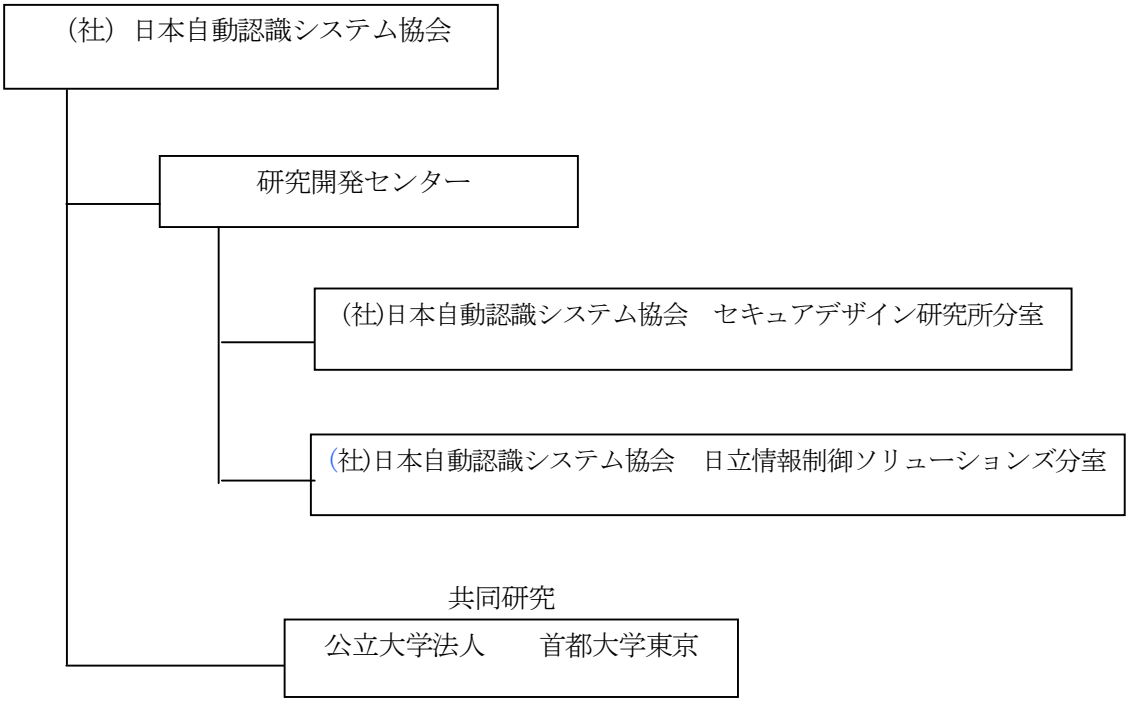
２．６　研究体制

（1）管理体制及び研究体制

1）管理体制



2）研究開発体制



（2）研究者名及び役職名

氏名	所　　属	役　割
栗田　寛久	(社)日本自動認識システム協会　研究員Ａ（セキュアデザイン研究所） ＩＳＯ／ＩＥＣ　ＪＴＣ１　ＳＣ３７　ＷＧ２ プロジェクトエディタ	推進研究者
高山　栄一	(社)日本自動認識システム協会　研究員Ｂ（セキュアデザイン研究所）	担当研究員
武野内伸子	(社)日本自動認識システム協会　研究員Ｃ（セキュアデザイン研究所）	ｿﾌﾄｳｪｱ担当
宇都宮康夫	(社)日本自動認識システム協会　研究員Ｄ （日立情報制御ソリューションズ）	推進研究者
柴田　夏彦	（社）日本自動認識システム協会　主任研究員	担当推進責任
中嶋　晴久	（社）日本自動認識システム協会　主任研究員	統括推進責任
柴田　彰	（社）日本自動認識システム協会　研究開発センター長	推進サポート

（3）経理担当者氏名及び機器装置等の管理責任者氏名並びに役職名

受託先　社団法人日本自動認識システム協会
（経理責任者）氏名　小池　勉　　所属　事務局　　事務局長
（管理責任者）氏名　大木　勝雄　　専務理事

２．７　成果の概要

（1）ISO/IEC 19784-1:2006 BioAPI２．０ 国際規格に準拠するバイオメトリックアプリケーションおよびフレームワームの製品について、その適合性の試験方法を標準化するため、関連規格の調査、解析を行った。その結果、ISO/IEC 24709-2 Conformance Testing for BioAPI: Part2 Test Assertion for BioAPI BSPにおいて　適合性の評価対象となるBioSPI関数9種類について用件定義が欠落していることが判明した。

（2）適合性試験用テストプログラムの製作のための基本仕様、および詳細仕様をまとめた。
（3）この仕様作成の段階において、フレームワークに対する標準化を優先すべしという国内ヴェンダーの要求が高まり、ISO/IEC JTC1 SC37 WG2において、エディターを介して日本からの提案として国際規格ISO/IEC 24709-3の開発の優先を承認させた。また、BioAPI準拠アプリケーションについては、改めて日本からのNP提案を行うということで各国の承認を取る事ができた。

（4）BioAPI2.0規格の根幹を成す構成要素であるフレームワークは、現在世界において1社のみが国際規格準拠製品として販売している。この会社との契約を成功させ、動作テストを開始した。テスト結果として基本的な動作には問題がなく、また当初予測された動作速度、高負荷に対する懸念事項についても問題がないことが確認できた。今後更なるテストにより、この結果の追検証をおこない国内企業への紹介を行う。
（5）BioAPI機能の解析が進み、国内においてBioAPI応用アプリケーションの開発を促進するベー

スができつつある。

（6）ＪＴＣ１から廃案テーマとなりつつあったが､今後のＢｉｏＡＰＩ関連適合試験について、プログラムの製作はじめ、国際標準化機構への日本からの貢献ができた。

２．７．１　適合性試験プログラムの開発（ＢｉｏＡＰＩフレームワークのテストプログラム)

・レファレンスフレームワーク入手・解析動作確認

レファレンスフレームワークのプログラムをＢｉｏｆｏｕｎｄｒｙ社から入手しプログラムおよび各種の設定パラメータを準備しシステムとしての動作が想定通りか確認を行った。また基本的な関数についてもその動作を確認し正しく機能しているかを確認した。

・プログラム仕様検討

テストされるプログラムとのインターフェースとしてＢＳＰ側およびアプリケーション側との関数の呼び出しインターフェースおよびコールバックインターフェース等で必要な機能についての仕様を検討する。検討する関数は約８０。

・プログラム基本設計

ＢｉｏＡＰＩの各関数の試験方法を定義した。

ＢｉｏＡＰＩ規格への適合性の判定基準を設定した。

CTSプログラムの制御構造、流れ。機能、性能の定義を行った。

・プログラム詳細設計

基本設計で定義された内容を基にXML　言語による仕様定義を行う。

プログラミング言語（C,C＋＋、およびＪＡＶＡ等）での開発との整合性を確認する。

２．７．２　ＩＳＯ／ＩＥＣ２４７０９関連国際規格の解析

・ＩＳＯ／ＩＥＣ１９７８４－１（ＢｉｏＡＰＩ）で定義されているフレームワーク仕様､ＢＳＰ仕様およびアプリケーションインターフェースの仕様の調査､解析を行ったBioAPI２．０フレームワーク準拠製品に対する適合性の要求仕様としてＢｉｏＡＰＩ２．０の基本仕様のまとめと共に、適合性試験に対する要求仕様を確認した。

・ＩＳＯ／ＩＥＣ２４７０９－１および２で定義されているＢｉｏＡＰＩ規格への適合性試験の検証方法の妥当性の調査､解析を行った。

この結果､２４７０９-２において１５のＢＳＰ関数に関する適合性評価試験の定義が欠落していることが判明した。

・これらの調査解析を通してフレームワークテスト用ＢＳＰの仕様､アプリケーション仕様の機能を確認するとともにＩＳＯ／ＩＥＣへのＰａｒｔ３､Ｐａｒｔ４だけでなく

Ｐａｒｔ１やＰａｒｔ２に対しても修正提案の必要性を見直し検討を行った。

２．７．３　ＩＳＯ／ＩＥＣ　ＳＣ３７国際会議および成果の公表

・ＤｏＤ／ＮＩＳＴ，ＢＣＣ（バイオメトリック・コンソーシアム・コンファレンス）
ＢＣＣの会場等を中心に米国国土安全保障省はじめ米国バイオメトリクス関係筋からの情報収集はじめ各国の適合評価試験のあり方を調査した。
ＯＳＳ　Ｎｏｋａｌｖａ社の担当者とＢｉｏＡＰＩ２．０準拠フレームワーク製品の入手交渉を行い契約の準備を開始することに合意が取れた。

・ＡＢＣ（アジア・バイメトリクス・コンソーシアム）
第１回シンガポール開催にてアジア諸国の適合評価試験の取り組み動向の調査をおこなった。

・ＳＣ３７　ベルリン会議およびイスラエル会議　にて

ベルリン会議にて、国内産業界の要請により、２４７０９－３の開発を早めるために日本からエディター就任を提案し受理された。
イスラエル会議では、ＪＴＣ１からの２４７０９－３，４に対する廃案警告に対して、国内産業界の要請により、２４７０９－３を最重点テーマとして優先的に開発をすすめ、ＩＳＯ／ＩＥＣ２４７０９－４は一時廃案とし、後日再度日本からＮＰ提案を行うことを提案し、承認された。ＩＳＯ／ＩＥＣ２４７０９－３　ベースドキュメントの提案を提案し各国に回覧することとなった。

２．７．４　　ＢｉｏＡＰＩを初めとするＳＣ３７の適合性評価の実運用方法の調査

日本におけるISO/IEC JTC１　SC37に関する適合性評価体制および体制構築の戦略的なシナリオを，日本の産業力強化の観点から検討した。
バイオメトリック市場は，2001年の米国の同時多発テロをきっかけに、日本では銀行カードのスキミング不正使用などにより，静脈認証技術の実用化、さらに2006年の電子パスポート発行に始まり、IC運転免許証、入国管理システムなどの社会公共システム市場が本格的に立ち上がっている。今後バイオメトリック技術がコモディティ化し、モバイルPCや端末などにおける本格導入が考えられる．この場合，利用者に代わって，中立的な第三者による評価検証が重要となる。
海外では，米国、欧州、韓国、シンガポール等の国々において、適合性評価試験に対するプロジェクト及び体制の整備が開始、実質的な適合性評価ノウハウを蓄積している。．一方日本では，これまで実施あるいは設置する機会はあったが，長期的な戦略のもとにプロジェクトが実施されていないため，ツールの開発や組織設置に関するノウハウが蓄積されていない問題がある．

適合性評価に関しては，第一者適合性から第二者，第三者適合性評価があり、どの評価体制をとるかは，測定する評価対象を考慮した検討が必要である．精度などの性能評価に関する適合性評価は，可能ならば，強いユーザによる第2者適合性が適切である．あるいは，評価方法が国際標準に準じて行われたか否か手順認証を行うことが適切であると判断する．一方，APIなどの仕様に関しては，評価すべきはプログラムインタフェース仕様のため，比較的客観的な評価が可能であり，第三者適合性評価が有効である．

アジアにおいては，静脈など日本独自の製品があるため，他国の評価機関を利用するということは，産業戦略上考えにくい，日本に適合性評価組織を設置することが適切である．今後、アジアにおける適合性に関するイニシアチブと，ツールの開発，運用ノウハウを蓄積するため，2009年に日本で開催予定されるＡｓｉａ　Ｂｉｏｍｅｔｒｉｃ　Ｃｏｎｓｏｒｔｉｕｍ（ＡＢＣ）において，適合性評価コンテストを実施することが有効と考える．

２．７．５　まとめ

現在、バイオメトリクス関連製品への国際規格が順次生まれている。今回のテーマはこれら製品開発に適用されるＩＳ化ルールに適合しているかどうかの試験方法をＩＳ化にて規定するテーマ活動である。このような評価試験についてはＩＳＯ／ＩＥＣ　ＪＴＣ１　ＳＣ３７ではWG２のＢｉｏＡＰＩが先行して進んでいるが、性能評価含めるとWG３，ＷＧ５でも動きが生まれている。

本テーマでは各国、各ベンダーで開発されたバイオメトリクス製品の相互互換性を保つための国際標準化ルール作りである。つまり、ＩＳ化ルールに則りバイオメトリクス関連製品が生まれつつあり、国を跨ぎ、システムを跨いでも運用が可能になる。今回の基準認証テーマでは平成１９年度を初年度とし平成２１年度まで、向こう３年がかりで日本としての適合試験評価の基礎を固めるプロジェクトである。

ＩＳＯ／ＩＥＣ　ＪＴＣ１　ＳＣ３７　WG２　BioAPIを３層に分け、そのうち最も重要な中層部の適合性試験規格がＩＳＯ／ＩＥＣ２４７０９－３である。これはもともと韓国提案であるが、この標準化活動についての国際エディタは日本の栗田氏が務めることになり、この分野における日本の動きが国際的に脚光を浴びている。続いてＩＳＯ／ＩＥＣ２４７０９－４テーマについても韓国提案であったが、登録後の期限超過にて一旦廃案となった。今後は日本の産業界にて議論しつつ、ＮＰ提案を提出してテーマを再開させていく予定である。

今年度の具体的活動については３章、４章に詳細がまとめられているように予定通り、１年目を終えた。簡単に振り返ってみると、当初先行していたＩＳＯ／ＩＥＣ２４７０９－４はWD２まで進めていたが、日本の産業界の中で中枢部であるＩＳＯ／ＩＥＣ２４７０９－３を先行させるべきとの意見より、Ｐａｒｔ３を優先し、ベルリン会議で国際合意を得た。以降、実際にＰａｒｔ３に関与するＢｉｏＡＰＩ、並びにＢｉｏＳＰＩ関数についての仕様の特定を行い、イスラエル会議にベースドキュメントとして提案、直後WDとし

て提案中である。併せて既にＩＳ化されている関連規格であるＩＳＯ／ＩＥＣ２４７０９－１、２やＩＳＯ／ＩＥＣ１９７８４－１についても日本語解釈し、適合評価試験要領全体の再見直しを行なった。結果的に今後既成ＩＳ化ドキュメントに対しても修正すべく提案を検討中である。

今年度に開催されたＩＳＯ／ＩＥＣ　ＪＴＣ１　ＳＣ３７のベルリン、イスラエル国際会議を皮切りに、日本として今後の国際会議の場面を活用して適合試験評価のＩＳ化を主体的に進めていく計画である。実際に机上にて関数仕様を決めてその妥当性を証明するためにも具体的なデモが必要である。ＢｉｏＡＰＩ関数４７種、ＢｉｏＳＰＩ関数３９種あり、その仕様を決め実際に各バイオメトリックシステムに適用する。その結果Ｙｅｓ　ＯＲ　Ｎｏ　の判定が生まれる。この判定が正しく稼動するかどうかの実感を国内外ＳＣ３７関係者間で共有することが必要である。

これらの国際的な共有化タイミングとしてAsia Biometric Consortium ２００９ ＩＮ ＴＯＫＹＯを絶好の機会と考えている。そのためには関数を主としたプログラムの正確性はもちろんのこと、国際会議などでのＰＲや勧誘、もちろん国内での本テーマ参画者の勧誘活動も見落としてはならない。

これらのデモによる実証検分を経て適合試験評価のあり方を進めていく一方で、日本におけるISO/IEC JTC１ SC37に関する適合性評価体制および体制構築の戦略的なシナリオを検討してきた。適合性評価に関しては，第一者適合性から第二，第三者適合性評価がある。国際的な、また広域的な活用の増大に伴いAPIなどの仕様に関しては，評価すべきはプログラムインタフェース仕様のため，比較的客観的な評価が可能であり，第三者適合性評価が有効である。

日本に適合性評価組織を設置するかどうかという点についてであるが、技術的対応等が必要であるがゆえに、日本の企業が海外へ適合試験のためにバイオメトリック製品を持参し出張することは非効率である。また、定型的な試験項目内容といえども具体的に関連技術情報を提供せざるを得ないことも考えられるため、国内設置が望ましいことは自明の理であるが、実用性や経済性との検討も必要である。

３　関連国際規格の調査及び解析

３．１　ISO／IEC19784－1:2006Information technology－Biometric application programming interface－Part 1:BioAPI specificationについて

３．１．１　はじめに

・複合認証はサポートしていない。
・構成要素の相互運用を可能とするようなAPIを定義している。
・BioAPI Frameworkはbiometric service provider(BSP)の構成要素を呼び出している。
・BioAPIの最下層には画像の収集、照合、データ保存のようなバイオメトリクスの基本機能を実行するBioAPI Unitが存在し、これはBSP内に含まれるかBioAPI Function Provider(BFP)として独立のユニットとして供給される。
・BioAPI Unitはシステムから動的に挿入、取り除くことができる。
・ＢioAPIの挿入、取り除きはイベントとして取り扱い、BSPとBioAPI Frameworkを通してアプリケーションへ通知される。
・BioAPI Unitを使用してバイオメトリクスの基本機能、登録、照合、識別及びデータ保存を利用することができる。
・バイオメトリックシステムに実装されたバイオメトリック構成要素の情報はbiometric component registry内に定義されている。

（１）　適用範囲

・APIとSPIを定義する。
・複数のベンダーから提供される構成要素間の相互運用性を用意する。
・バイオメトリックアプリケーションとBSPのセキュリティーは適用範囲外である。
・相互運用性とバイオメトリックシステムのパフォーマンス間のトレードオフは適用範囲外である。

３．１．２　ＢｉｏＡＰＩの構造

（１） 仕様概要（付録C）

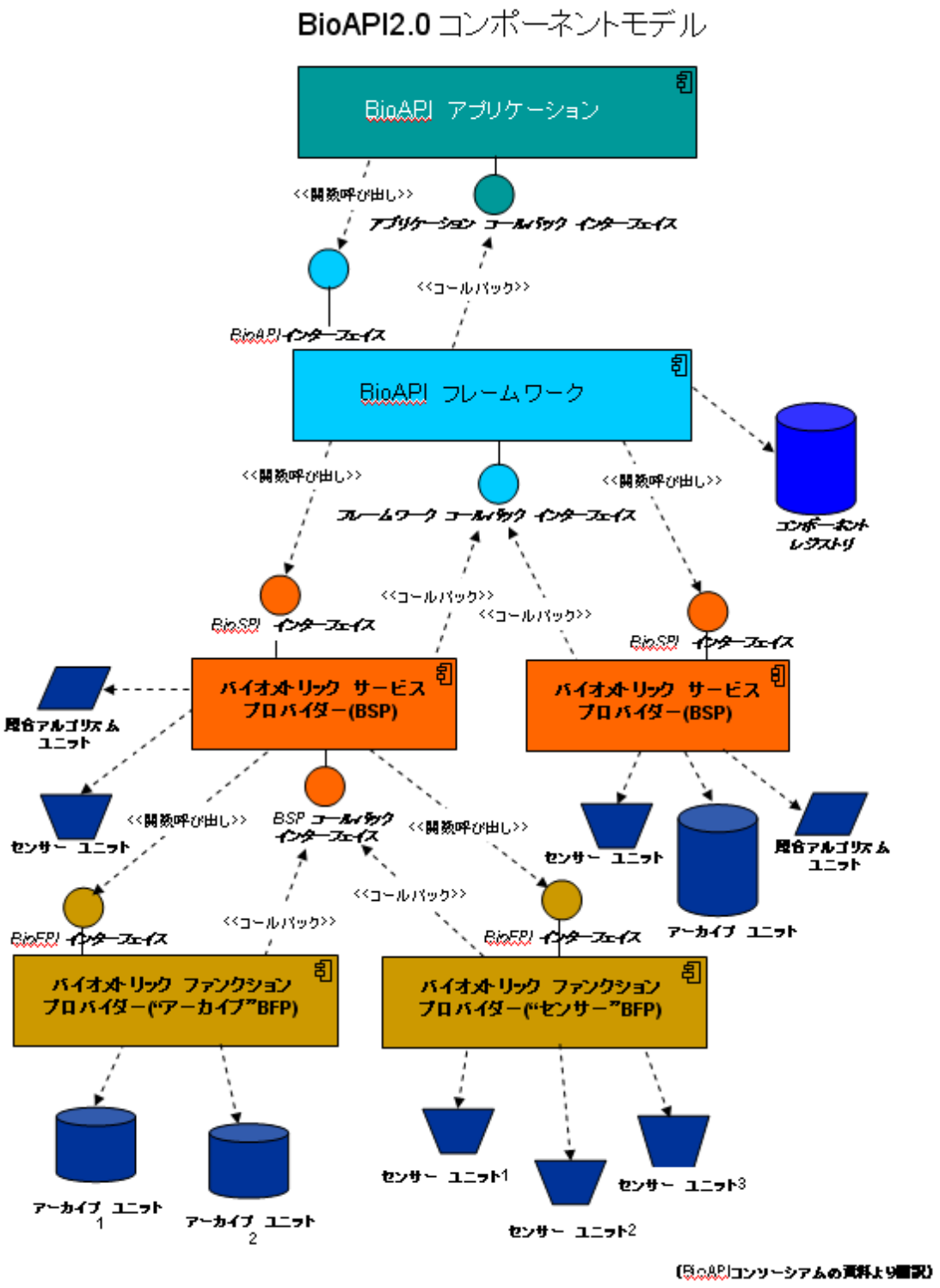
1) アプリケーションレベルのAPI

・アプリケーションレベルはバイオメトリックス機能を使ったアプリケーションの最上位のレベルである。
・アプリケーションレベルのAPIはバイオメトリック認証に必要な全ての機能を含む。
・従って大集団に対する識別（1:N照合）機能は特別なBSPとして提供されるためオプション機能である。
・BioAPIはBSPの開発者及びアプリケーションの開発者が使用するために設計されている。
・バイオメトリック技術を用いたソフトウェアのインテグレーションにおいてベンダー固有の実装方式、製品、装置及びセンサー等の特異性及び複雑性を隠蔽し、カプセル化して効率化する。
・このインテグレーションにおいてはBioAPIインターフェイス及びBSPが提供する標準インターフェイスを通してバイオメトリックスの基本メカニズムをアプリケーションからアクセスするこ

とが可能となる。

・理論的にはBioAPI規格に適合したBSPを提供するベンダーの製品はこのBioAPIの定義によって開発されたアプリケーション内において最低限BioAPIの基本レベルの機能を実行することができる。

・本仕様においては複合認証方式をサポートするように設計されている。



2） バイオメトリック技術

・BioAPIの基本モデルはバイオメトリック技術を用いた全てのタイプについて共通である。

・登録に使用されるアルゴリズムはベンダー固有でもいい。

・最初に生成されたテンプレートはアプリケーションにより参照テンプレートとして保存される（アーカイブBioAPI Unit上にBSPによって保存される場合もある）。

・バイオメトリック認証には1:1照合(Verification)と1:N照合(識別、Identification)がある。
・図３－１に可能な実装アーキテクチャが表示されている。
・図に表示されている各段階、キャプチャー、処理、照合は１つのアプリケーション又はBSPに対応させることができる。
・全てのステップを１つのデバイスの中で実行させることもできる。

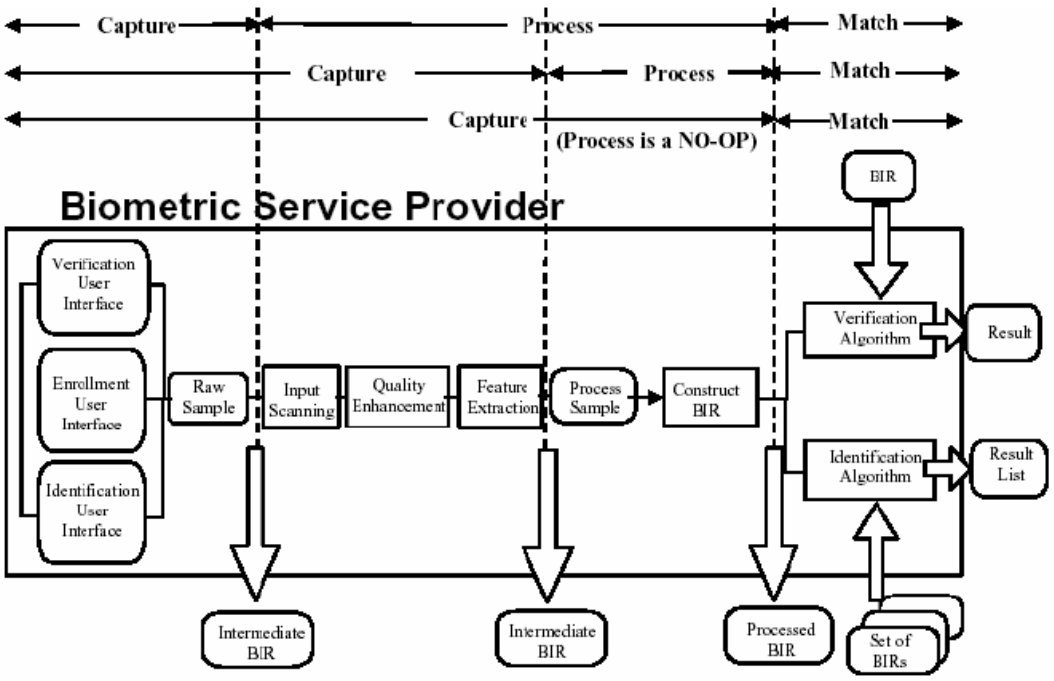


Figure C.1 — Possible Implementation Architectures

図３－１　実装アーキテクチャ

３） バイオメトリック関数

・BioAPIには３つの主要な高位の抽象化された関数がある。

①登録

センサーユニットからサンプルが収集され、テンプレートを作成する利用可能な形に処理され、アプリケーションに戻る。

②照合

１つ又は複数のサンプルが収集され、利用可能な形に処理される。そして参照テンプレートとマッチングされる。比較の結果が戻される。

③識別

１つ又は複数のサンプルが収集され、利用可能な形に処理され、テンプレートの集合とマッチングされる。サンプルがその集合の中の最上位の候補と比較されて、どのくらい似ているかということを示すリストが戻される。

・図３－１に示すように、生のサンプルから収集されたバイオメトリックデータを１つのテンプレートに対して照合するための処理は、多くの段階で実施されることもある。
・BioAPIは、その実装者に対して、その中に含まれるプロセスの処理が様々な段階で処理されるような最大の自由度を与え、そしてセンサーが付いているクライアントマシンとサーバの間でその処理が共有できることをカウントしている。
・BioAPIはまた内部でバイオメトリックの処理が全て実行されるような自己完結型の装置も許容している。
・サーバ側で照合と処理プロセスが行われる長所。

ア）アルゴリズムがより安全な環境で実行される。（例えば、ファイヤーウォールの後のアクセスコントロール環境において）

イ）クライアント側ではアルゴリズムを効率的に実行する能力が不足している。

ウ）ユーザのデータベース（バイオメトリクスによって保護されている資源）がサーバ上に置かれている。

エ）非常に大きな集団に対する識別はサーバ上でのみ合理的に実行できる。

(a)　基本関数

・Capture、Process、Match（Verify MatchとIdentify Match）、およびCreate Templateの４つの基本関数は上位レベルの抽象化された関数と同じ役割を果たす。

(ｱ)　画像収集（キャプチャ）

・画像収集はクライアント側で常に実行される（「機能がサポートされていません」という場合は除く）。
・Capture関数はサンプルに対してBIRの作成を完了するために、照合や識別に対して適するような形に合わせる。
・処理が不完全な場合にはCapture関数は”中間”BIRを戻し、Process関数（又はCreateTemplate関数）が要求される。
・プロセスが完了した場合には、Capture関数は”処理済”BIRを戻し、Process関数が呼び出される必要がないことを示す。
・アプリケーションはサンプルに対して目的となる処理を定義する。
・この目的は作成されたBIRのヘッダーに記録されている。

(ｲ) 処　理（プロセス）

・画像処理アルゴリズムの実行はクライアント側又はサーバ側、どちらでも可能である。

・Process関数は照合又は識別の目的のために必要なサンプルの処理方法を提供する。

・Process関数は常に入力として”中間”BIRを取り、その目的のために最終的な形にバイオメトリックデータの処理を完了することができる。

・クライアント側では、”処理済”BIRか”中間”BIRが戻される。従ってProcess関数はサーバ上でも呼び出される必要がある。

・サーバ上では処理が常に完了しなければならない。そして、常に”処理済”BIRが戻される。

・アプリケーションは常にサーバ上で処理を行うことを選択できるが、ネットワークの負荷を低減させるために、クライアント側でProcess関数を呼ぶこともできる。

注) “処理済”BIRは常に”中間”BIRより小さい。その度合いは技術に依存し、またCapture関数によってどのくらいの処理が実行されたかに依存する。

(ｳ) 照合（Verify Match関数とIdentify Match関数）

・照合のマッチング（Verify Match）は１つのテンプレートと”処理済”BIRの間の比較として実行される。

・識別のマッチング（Identify Match）はテンプレートの集合と”処理済”BIRの間の比較として実行される。

・Identify Match関数はオプションである。

(ｴ) テンプレートの生成

・CreateTemplate関数は登録用のテンプレートを作成する。

・CreateTemplate関数は入力として常に”中間”BIRを取り、BIRの中に目的の定義としてBioAPI_PURPOSE_ENROLL、BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY、BioAPI_PURPOSE_ENROLL_FOR_IDENTIFICATION_ONLYを記録する。

・オプションとして、”中間”BIRの中に含まれる新しいバイオメトリックサンプルを用いて古いテンプレートを新しいテンプレートに置き換えたり更新したりすることができる。

・BSPはアプリケーションに”ペイロード”の保存を許す（C.5節参照）。

・図３－２にクライアントサーバ間での基本関数の流れを示してある。

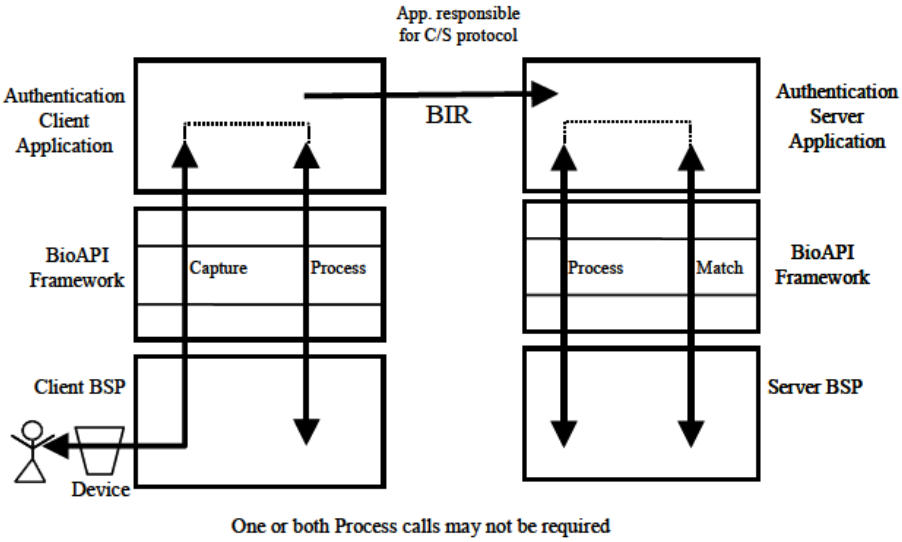


Figure C.2 — Example Client/Server Implementation Using Primitive Functions

図３－２　クライアントサーバ型の実装形態

注）クライアントサーバ型の実装形態はISO/IEC 24708 Biometric Interworking ProtocolにBioAPI Framework間の交信として定義されており、これを使用することもできる。

（２）　BioAPI API/SPI構造モデル

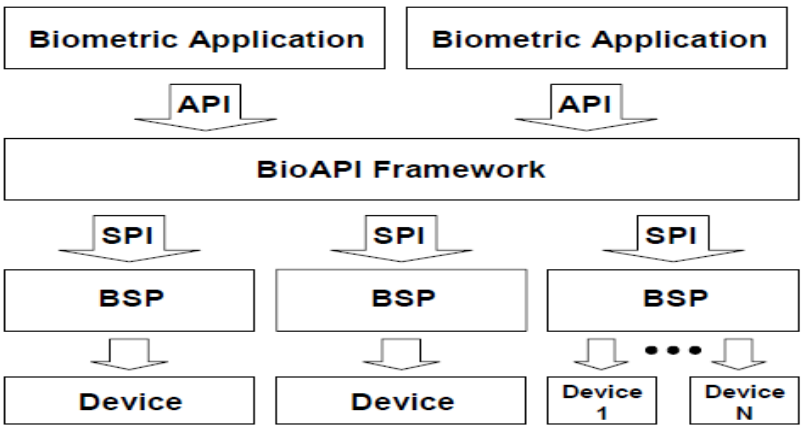


Figure 1 — BioAPI API/SPI Model

図３－３　BioAPI API/SPI構造モデル

・APIはBioAPI Frameworkとバイオメトリックアプリケーションの間のインターフェースを定義する。
・アプリケーションはAPI定義の中の関数を呼び出すように書かれている。

・BioAPI　FrameworkはAPI定義の中の関数をサポートする。
・SPIはBioAPI FrameworkとBSPの間のインターフェースを用意する。
・BioAPI FrameworkはSPI定義の中の関数を呼び出す。
・BioAPI FrameworkはBSPのマネージメントを用意し、適当なBSPを取り扱うためのSPI関数をAPI関数からコールする。
・１つのアプリケーションは一度に複数のBSPをロードすることができる。
・１つのBSPは一度に複数のアプリケーションにアタッチすることができる。
・次のような状態がサポートされている。

a)　１つの BioAPI Framework とそれに付随するコンポーネントレジストリ

b)　BioAPI Framework と相互作用し複数の同時実行バイオメトリックアプリケーションのダイナミックな実行と終了

c)　BioAPI Framework と相互作用する複数の BSP のダイナミックインストールと取り外し

d)　BSP が管理する BioAPI Unit のダイナミック結合及び取り外しに関連したイベントの発生信号を BSP から BioAPI　Framework へ発信する

（３）　BioAPI BSP構造モデル

・BioAPI Unitはバイオメトリック装置の抽象化であり、BSPによってアプリケーションに提供される機能をサポートする基本ブロックである。
・BioAPI Unitの分類

- Sensor unit
- Archive unit
- Matching algorithm unit
- Processing algorithm unit

・BioAPI Unitのマネージメント方式としては２つあり、

(a) BSP の内部に含まれ、BSP が直接コントロールする

(b) BSP が BioAPI Function Provider（BFP）を通して間接的にコントロールする。

・BFPは複数の分野のBioAPI Unitを管理することができる。しかし、１アプリケーションにおいて各セッションについては１個のみである。
・BFPの分類

- Sensor BFP
- Archive BFP
- Matching algorithm BFP
- Processing algorithm BFP

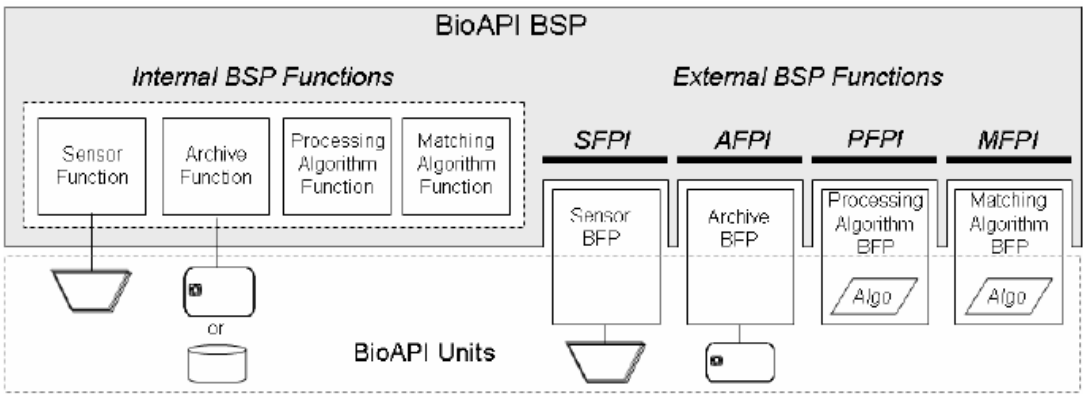


Figure 2 — Illustration of BSP architecture

図３－４　BioAPI BSP構造モデル

・BioAPI関数とそれに関連したSPI関数（又はFPI関数）があり、これらによりアプリケーションはBioAPI Unitとの間で情報交換する。これは追加仕様である。
・これらの制御関数のパラメータは規格化されていない。
・BSPのアタッチセッションが設立されたときには、各カテゴリーの中で１つのBioAPI Unitが選択されます。

(4)　コンポーネントレジストリ

・コンポーネントレジストリはインストールされたBSPとBFPの情報を含んでいる。
・BioAPIモデルでは、１つのBioAPI Frameworkとそれに関連する１つのコンポーネントレジストリが存在する。
・BioAPIの要求仕様として、１つのコンピュータシステム内に存在する異なったバイオメトリックシステムは、お互いに独立である（相互作用、及び干渉がない）。
・アプリケーションはBioAPI Framework関数によってコンポーネントレジストリから次の情報を得ることができる。

a)　BioAPI Framework 自身の情報

b)　実装済み BSP の詳細情報

c)　実装済み BFP の詳細情報

※BSPは実装済みBFPの情報をコールバック関数を通して得る。
・アプリケーションが得ることができる情報（特定のBSPからSPIによって得られる）。

a)　BSP がサポートする全ての実装済み BFP の詳細情報

b)　BSP によりアクセスすることができる挿入状態の全ての BioAPI Unit の詳細情報

・コンポーネントレジストリ情報の入手タイミング。

a) framework 自身の情報は BioAPI_Init 関数実行後、任意のタイミングで可能である

b) 実装済み BSP の詳細情報も同じ

c) 実装済み BFP の詳細情報も同じ

d) 各 BSP がサポートしている実装済み BFP の詳細情報は、BioAPI_BSPLoad 関数実行後、任意のタイミングで可能である

※1: １つのアプリケーションにおいて複数のBSPの同時ロードが可能。

※2: これはBioAPI_BSPLoad関数が実行された時、BSPは使用可能な全てのBFPをロードする為にFPI関数を使用する。

(5) BSPとBFPのインストールと取り外し

・BSP、BFPをインストールするためには、UUIDが必要である。

・BSPのインストールはBioAPI Framework関数の呼び出しにより行われる。

・BSPのインストールは通常のバイオメトリックアプリケーションではなく、特別なインストールプログラムが必要である。

・BSPのインストールは実装するコンピュータシステムの機能に依存したインストールプログラムを使用する。

・インストールするBFPの情報はコールバック機能によってFrameworkから得ることができる。

・インストールされたBSP（現在稼動中、又は停止中に関わらず）へ新しくインストールされたBFP、又は存在するBFPの取り外しに関する情報を伝達する機能はない。

・BSPがどのBFPをインストールするかを決定する方法としては、BFP Enumeration Handler コールバック関数を使用してのみ可能である。

・BioAPI FrameworkへBSPがインストールされたという情報を伝達するために、アプリケーションが呼び出すことができるBioAPIのAPI関数がある。

・BioAPI Frameworkはコンポーネントレジストリを更新する。

・BFPの取り外しの効果についてBSPへ情報を伝えるような標準関数はない。

※BFPの取り外し方法については本BioAPI規格外である。

(6) BSPのロードとBioAPI Unitの取り付け

・バイオメトリックアプリケーションの動作の流れ。

a) BioAPI Framework のイニシャライズ

b) BSP のロード

c) BSP に関連した BioAPI Unit のアタッチ

d) BSP のロード、及び BioAPI Unit のアタッチに関連したその他の BioAPI 関数の呼び出し

・BSPをロードするとBioAPI Unitの完全な情報を得ることができる。
・BioAPI_DONT_CAREを使用したBioAPI Unitのデフォルト選定。
・１つのアプリケーションが１つのBSPをアタッチした時にBSPによってBioAPI Unitの特定のカテゴリーから特定のBioAPI Unitを選定し、アタッチすることである。
※１つの与えられたカテゴリーの中から特定のBioAPI Unit選定とBioAPI_DONT_CAREというコマンドを使ってデフォルトのBioAPI Unitを選定する間の違いは、どのBioAPI Unitを選択するかの決定をBSPが行うことである。
・BioAPI Unitの挿入イベントの通達にはBioAPI Unitのスキーマが含まれる。
・BioAPI_BSPAttach関数の実行にあたり、アプリケーションはBioAPI Unitを選定しなければならない。
・BSPの取り外しに関するイベント通達に含まれる情報。

a) BioAPI Unit ID

b) Event type(remove)

c) Callback context

・BioAPI Unitの取り外しのイベントが発生した時、アプリケーションはBioAPI_BSPDetachとBioAPI_GetBIRFromHandleの２つの関数以外は実行しない。

(7) BioAPI Unitの制御

・BioAPIのAPI関数、SPI関数、そしてFPI関数によりアプリケーションはBioAPI UnitをBSPを通して制御することができる。

(8) BIRの構造と処理方法

1) BIR構造

・API及びSPI関数の中で表示されるBioAPI BIRは、本規格書７．４節で定義されたデータ構造を使用する。

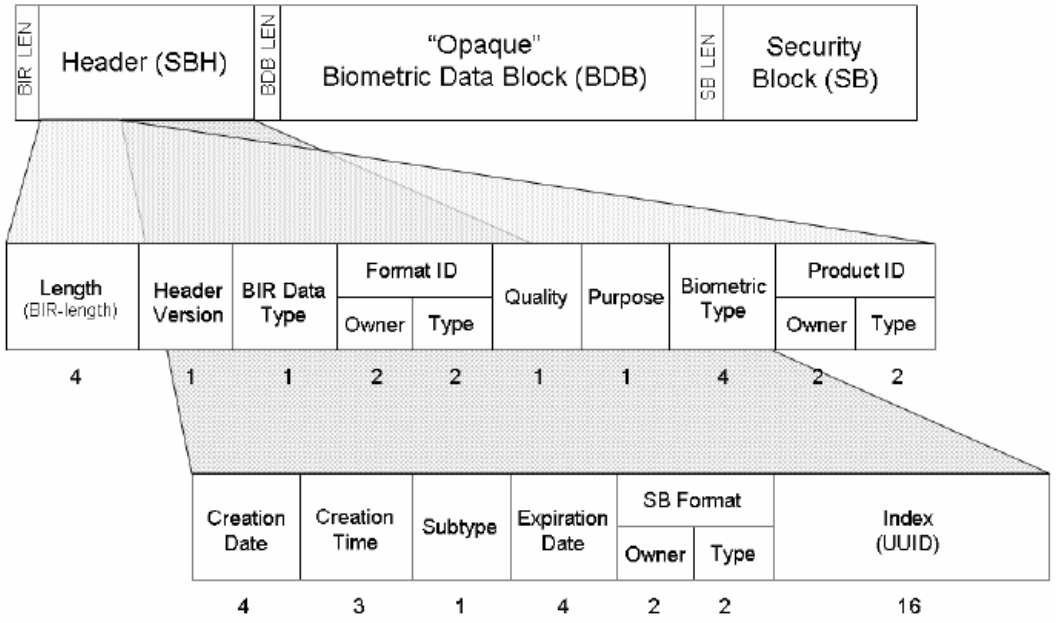


Figure 3 — Biometric information record (BIR)

図３－５　BioAPI BIR

・Standard Biometric Header(SBH)は次に続くBDBの内容に関する情報を含む。
・BDBはバイオメトリックサンプルのデータを含み、SBHのFormat ID フィールドでフォーマットが定義されている。固有フォーマットと標準フォーマットがある。
※標準BDBフォーマットはISO/IEC 19794で定義されている。
・セキュリティーブロック(SB)はオプションであり（存在しない場合はゼロレングスフィールド指定される）、BIRの暗号処理と署名に関するパラメータを含んでいる。
・SBのフォーマットはSBHのSB Format フィールドで定義されている。
・BIR Data TypeはBIRが署名付きかBDBが暗号化されているかどうかを指定する。

2)　BIRのデータ処理
・BSPが新しいBIRを生成する場合、ハンドルが戻される。

３．１．３　ＢｉｏＡＰＩのデータ型とマクロについて

本規格　ISO/IEC19784-1 の第７節BioAPI　Types and macrosを参照ねがいます。

３．１．４　ＢｉｏＡＰＩ関数仕様

本規格　ISO/IEC19784-1 の第8節BioAPI functionsを参照ねがいます。

３．１．５　ＢｉｏＡＰＩバイオメトリックサービスプロバイダー（ＢＳＰ）について

ＢｉｏＡＰＩ関数とＢｉｏＳＰＩ関数の処理フローに関しては本規格ISO/IEC

19784-1の９節を参照願います。

３．１．６　適合性評価について

（1）　一般事項

・ISO/IEC19784の本規格部分に対する適合性評価は次の３つのクラスに分けられる。

- BioAPI適合生体認証アプリケーション
- BioAPI適合BioAPIフレームワーク
- 以下のサブクラスの分類の一つに準拠するBioAPI適合BSP
  - ○　BioAPI準拠照合BSP
  - ○　BioAPI準拠識別BSP
  - ○　BioAPI準拠キャプチャBSP
  - ○　BioAPI準拠照合エンジン
  - ○　BioAPI準拠識別エンジン

・生体認証アプリケーション、BioAPIフレームワーク, BSPは（２）、（３）、（４）にてそれぞれ定義されている。

注記）BFPの適合性については、ISO/IEC19784の本規格書内では定義されていない。しかし、今後策定される本規格の複数部分の中で定義される。

・BioAPIユニットへのインターフェースは標準化されていない。

⑵　BioAPI準拠バイオメトリックアプリケーション

・BioAPI仕様への準拠を宣言するためには、生体認証アプリケーションは各BioAPI関数呼び出しを使用し、本規格で定義された動作と一致する関数呼び出しを行わなければならない。すなわち、全ての入力パラメータが正当であり存在しなければはならない。

・呼び出される必須な関数の最小の数というもの存在しない。しかし、生体認証アプリケーションは、これらの関数に定められた呼び出し方式に準拠しなければならない。

⑶　BioAPI準拠フレームワーク

・BioAPIフレームワーク・コンポーネントは、BioAPI準拠アプリケーションとBSPの間に存在する基本構造部となるミドルウェアである。以下に示す一般的な目的のために提供される。

a) BSPのロード/取り付け（アタッチ）
b) BSPとBFPの管理
c) コンポーネント・レジストリーのメンテナンスと管理
d) 本規格書８節にて定義されたAPI関数とこれに対応するSPI関数の間の解釈を実行する透過形の関数呼び出し
e) BSPからのイベント発生の通知処理と、これらのイベント通知をBSPをロードしたアプリケーション内の（可能性としては、複数の）イベントハンドラーへ送信すること

f) BioAPIコンポーネントのインストールと取り外し（デインストール）に関連したAPI関数呼び出しを行うと共に、コンポーネント・レジストリーの関連する内容を更新すること。

g) インストールしたBFPに関するBSPからの問い合わせをサポートすること

・BioAPI定義に準拠していることを主張するためには、BioAPIフレームワークは：

a) 本規格書８．１で定義されたコンポーネント・レジストリの機能と処理が用意されること。

b) 本規格書１０節に従ったコンポーネント・レジストリーのサービスが用意されること。

c) 本規格書８節で定義されたAPI関数とこれに関連する１１節で定義されているSPI関数との間の翻訳変換を実行する。

d) 上記a)からc)の項目が実装された場合に、本規格書７節で定義されたデータ構造と１１節で定義されたエラーコードに準拠すること。

e) 本規格書７．２８、８．３および９．２節で定義されているイベント通知とコールバックを処理すること。

・適合BioAPIフレームワークは本規格内で定義された全てのオプションをサポートすることが要求される。なせならば、これらの任意のオプションを装備するBSPとアプリケーションへのサービスをこのフレームワークは用意するからである。適合フレームワーク機能の一部分をサポートするものは定義されない。

(4) BioAPI適合BSP

・BioAPI仕様に適合すると主張するためには、BSPは、本規格書９節で定義されたSPIにしたがって、下記に定義されたように適合サブクラスの必須機能を装備しなければならない。BSPは本規格書Ａ．１．１節定義された適合サブクラスの一つに適合していると宣言できる。

・BSPは全ての正当な入力パラメータと戻り値の出力を受理しなければならない。オプションの戻り値と機能は適合性の公表には必要ない; しかし、本規格で定義された仕様を満たすために、実装された任意のオプションとしての関数とパラメータが装備されなければならない。追加としてのパラメータは必要ではない。

・BSPのインストールプロセスでは、コンポーネント・レジストリーに登録された全ての集団について処理が実行されなければならない。

・BSPは特定のバージョンとBSP製品に対する正当な固有UUIDを保持しなければならない。
注記）UUIDは自己生成（ISO9834-8参照）であり、同じBSP製品/バージョンがインストールされている複合システムでは同じでなけらばならない。（必須ではない）

・BSPにより生成されたBIRは、本規格書７節で定義されたデータ構造（これらはBioAPI BIRでなければならない）に適合しなければならない。BSPは登録されたフォーマット・オーナーと関連した正当なフォーマット・タイプを含むBioAPI_BIRデータをのみ戻す。（本規格書７．６節と付録Ｂ参照）

・BSPは本規格書１１節で定義されたエラー処理を実行しなければならない。

・全てのBSPは基本的なコンポーネント管理（本規格書９．３．１）、ハンドル（９．３．２節）、イベント（９．３．７節）およびユーティリティ（９．３．３　（２））をサポートしなけれ

ばならない。コールバック（９．３．３（２）節）、データベース（９．３．５節）及びBioAPIユニット（９．３．６節）動作はオプションである。

・次の表は、BSPのサブクラスに対する適合性の要求仕様をまとめたものである。詳細は次の小節に記載されている。Ａ４．６節では、オプション機能についての適合性に関して提示されている。

表３－１ －BSP適合性評価サブクラス

関数	照合 BSP	識別 BSP	キャプチャ BSP	照合エンジン	識別エンジン
コンポーネント管理関数					
BioSPI_BSPLoad	X	X	X	X	X
BioSPI_BSPUnload	X	X	X	X	X
BioSPI_BSPAttach	X	X	X	X	X
BioSPI_BSPDetach	X	X	X	X	X
BioSPI_QueryUnits	X	X	X		
BioSPI_QueryUnits					
BioSPI_QueryBFPs					
BioSPI_ControlUnit					
ハンドル関数					
BioSPI_FreeBIRHandle	X	X	X	X	X
BioAPI_GetBIRFrom Handle	X	X	X	X	X
BioSPI_getheaderFromHandle	X	X	X	X	X
コールバック及びイベント関数					
BioSPI_EnableEvent					
BioSPI_SetGUICallbacks	X	X	X	X	X
生体認証関数					
BioSPI_Captute					
BioSPI_CreateTemplate			X		
BioSPI_Process				X	X
BioSPI_ProcessWithAuxBIR				X	X
BioSPI_VerifyMatch					
BioSPI_IdentifyMatch				X	X
BioSPI_Enroll					X
BioSPI_Verify	X	X			
BioSPI_Identify	X	X			
BioSPI_Import		X			
BioSPI_PresetIdentifyPopulation					
データベース関数					
BioSPI_DbOpen					
BioSPI_DbClose					

BioSPI_DbFreeMarker					
BioSPI_DbStoreBIR					
BioSPI_DBGetBIR					
BioSPI_DbGetNextBIR					
BioSPI_DbDeleteBIR					
BioAPIユニット関数					
BioSPI_SetPowerMode					
BioSPI_SetIndicatorStatus					
BioSPI_GetIndicatorStatus					
BioSPI_CaliblateSensor					
ユーティリティ関数					
BioSPI_Cancel	X	X	X	X	X
BioSPI_Free	X	X	X	X	X

1) BioAPI適合照合BSP

・照合BSPは１対１照合（認証）を実行する機能をもったBSPである。１対N識別は実行しない。
注記）１対複数（１から少数の複数）照合は１対１照合の繰り返しとしてサポートされる

・BioAPI準拠の照合BSPは以下の生体認証関数をサポートしなければならない。

- BioSPI_Verify
- BioSPI_Enroll

(a) BioSPI_Verify

・最近似値のみが、より良いサポートされたMaxFMRRequestedとして提供されなければならない。しかし、BSPはそのサポートされている値（FMRAchieved）を戻り値として提供しなければならない。（そのBSPは粗いスコア値が実装されているかどうかをコンポーネントレジストリ内に提示しなければならない。）
注記）閾値とスコア化に関して、FMRの使用についての説明がある本規格書C．４節を参照のこと。

・Acceptanceとサブタイプの利用はオプションである。

・入力としてReferenceTemplateに関連するペイロードがある場合、および照合スコア値がFMRAchievedの値を十分に超える場合は、ペイロードの戻り値が必要である。（BSPはコンポーネントレジストリー内にペイロードを戻り値として与えるために必要な最小FMRを記録しなければならない。）

・AdaptedBIRの戻り値はオプションである。生データ（AuditData）はオプションである。

・全てのBSPは照合動作におけるキャプチャ部分に関連する任意の必要なユーザーインターフェースが（デフォルトのユーザーインターフェースとして）用意されていなければならない。

・もし、BSPが複数の出力BIRデータフォーマットをサポートする場合（コンポーネントレジストリー内のBSPスキーマに提示されているように）、この関数はグラフィック・ユーザーインター

フェースに対するアプリケーションの管理機能はオプションである。

(b) BioSPI_Enroll

・Purposeの値BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLYが受理されなければならない。もし、異なったPurpose値が要求され、それがサポートされない値である場合、BioAPI_RETURNとしてエラー条件が設定される。
・Payloadの受理はオプションである。（すなわち、もしペイロードが供給された場合、BSPがPayload Carryをサポートしない時には受理される必要はない）
・AcceptanceとSubtypeはオプションである。
・もし、BSPが複数の出力BIRデータフォーマットをサポートする場合（コンポーネントレジストリ内のBSPスキーマに提示されているように）、BSPはOutputFormat入力の受理とそのフォーマットでのNewTemplateが戻されなければならない。
・ReferenceTemplate入力の使用は、用意されている場合、NewTemplate出力を生成するが、オプションである。
・生データ(AuditData）が戻り値として与えられることはオプションである。
・全てのBSPは登録（Enroll）動作におけるキャプチャ部分に関連する任意の必要なユーザーインターフェースが（デフォルトのユーザーインターフェースとして）用意されていなければならない。

２）BioAPI準拠識別BSP

・識別BSP１対１の一致判定（照合）と同時には１対N 識別の一致判定を実行する機能を持ったものである。BioAPI適合識別BSPは次の生体認証関数をサポートしなければならない。

- BioSPI_Verify
- BioSPI_Identify
- BioSPI_Enroll

(a) BioSPI_Verify

・最も近い値のみ、より良いサポート値としてMaxFMRRequestedがサポートされなければならない：しかし、BSPはそのサポートされている値（FMRAchieved）を戻さなければならない。
注記）スコアの算出と閾値に対するFMRの使用方法の説明としてC．４節を参照すること
・AcceptanceとSubtypeの使用はオプションである。
・Payloadの戻り価は、入力ReferenceTemplate`に連携されており、スコアがFMRAchievedの価を十分に超えている場合には必要である。（BSPは、コンポーネントレジストリーの中にペイロードを戻り価として出力するために必要なFMRの最低値を入れておかなければならない。）
・AdaptedBIRの戻り値はオプションである。生データ(AuditData)の戻り値はオプションである。
・デフォルトとして、全てのBSPはVerifyの動作の一部分であるセンサーからのデータ収集部において何らかの連携するGUIを用意しなければならない。しかし、そのGUIに対するアプリケー

ションによる制御はオプションである。

(b) BioSPI_Identify

・より良くサポートされているMaxFMRRequestedの値に最も近い値が提供されなければならない。戻り値として、一致する候補Candidatesが戻されなければならない；しかし、BSPは、 次に最も近い値/増加分をFMRAchievedのフィールド値として戻り値を送ることが出来る。（BSPはコンポーネントレジストリー内に粗いスコア値の使用機能が装備されているかどうかを提示しなければならない。）
注記）本規格書C．４節のスコアー処理と閾値処理に関する説明を参照すること。
・AcceptanceとSubtypeの使用はオプションである。
・Binningはオプションである。
・生データ（AuditData）の戻り値はオプションである。
・デフォルトとして、全てのBSPはIdentifyの動作の一部分であるセンサーからのデータの収集部において何らかの連携するGUIを用意しなければならない。しかし、そのGUIに対するアプリケーションによる制御はオプションである。

（ｃ） BioSPI_Enroll

・Purposeの値として、BioAPI_PURPOSE_ENROLL、BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY、及びBioAPI_PURPOSE_ENROLL_FOR_IDENTIFICATION_ONLYのみが戻り値として受け取り可能である。もし、目的としてこのPurposeの値にサポートされていない値が設定された場合は、BioAPI_RETURNとしてエラーの状態を表す値が設定されなければならない。
・AcceptanceとSubtypeの使用はオプションである。
・もしBSPが複数のBIRデータフォーマット出力をサポートする場合は（コンポーネント・レジストリー内のBSPスキーマに指定されているように）、OutputFormat入力とNewTemplateの戻り値が受け取られなければならない。
・入力としてReferenceTemplateが用いられる場合、それを使用して出力としてのNewTemplateを生成することはオプションである。
・生データ（AuditData）の戻り値はオプションである。
・デフォルトとして、全てのBSPはEnroll（登録）動作の一部分であるセンサーからのデータの収集部において何らかの連携するGUIを用意しなければならない。しかし、そのGUIに対するアプリケーションによる制御はオプションである。

３） BioAPI準拠データ採集BSP

・BioAPIデータ採集BSPは、一つまたは複数の生体認証センサーに対するインターフェースを用意するBSPであり、他のBSP（生体認証エンジンのようなBSP、本規格書Ａ．４．４及びＡ．５参照）によって使用される中間BIRを戻り値として提供する。この型のBSPは、採集したデータの

処理や照合を実行する機能を備えていない。１つのBioAPI準拠データ採集BSPは以下のバイオメト
リック関数をサポートしなければならない。

- BioSPI_Capture

(a)　BioSPI_Capture

・戻り値としての生データ（AuditData）は、オプションである。
・AcceptanceおよびSubtypeの使用はオプションである。
・もしBSPが複数のBIRデータフォーマット出力をサポートする場合は（コンポーネント・レジストリー内のBSPスキーマに指定されているように）、OutputFormat入力とBIR中にNewTemplateの戻り値が受け取られなければならない。
・デフォルトとして、全てのBSPはBioSPI_Captureと何らかの連携するGUIを用意しなければならない。しかし、そのGUIに対するアプリケーションによる制御はオプションである。

４） BioAPI準拠照合エンジン

・BioAPI準拠照合エンジンは、生体認証のデータ処理と１対１照合のアルゴリズムを含むBSPである。しかし、これは、生体認証データのセンサーからの採集は含まない。これらは、典型として、センサーからデータ採集動作を実行する（データ採集または完全なBSPとして）他のBSPと連携して使用される。BioAPI準拠照合エンジンは次の生体認関数をサポートしなければならない。

- BioSPI_CreateTemplate
- BioSPI_Process
- BioSPI_VerifyMatch

(a)　BioSPI_CreateTemplate

・Purposeの値として、BioAPI_ PURPOSE_ENROLL_FOR_VERIFICATION_ONLYのみが受理される。他のサポートされない目的が設定された場合は、BioAPI_RETURNの値にエラーの状態を示す値がセットされる。
・もしBSPが複数のBIRデータフォーマット出力をサポートする場合は（コンポーネント・レジストリー内のBSPスキーマに指定されているように）、OutputFormat入力とBIR中にNewTemplateの戻り値が受け取られなければならない。
・Payloadの受理はオプションである。　テンプレートの更新（ReferenceTemplateの入力による）はオプションである。

(b)　BioSPI_Process

・Purposeの値としてBioAPI_PURPOSE_VERIFYと共に入力CapturedBIRのみが受理されなければならない。他の目的の値が定義された場合は、BioAPI_RETURNの戻り値にエラー状態を示す値が設定されなければならない。

・もしBSPが複数のBIRデータフォーマット出力をサポートする場合は（コンポーネント・レジストリー内のBSPスキーマに指定されているように）、OutputFormat入力とBIR中にProcessedBIRの戻り値が受け取られなければならない。

(c) BioSPI_VerifyMatch

・Purposeの値、BioAPI_PURPOSE_VERIFYと共に入力BIR（ProcessedBIR）、およびPurposeの値、BioAPI_PURPOSE_ENROLLまたはBioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLYと共に（ReferenceTemplate）のみが受理されなければならない。もし他の目的が設定された場合は、BioAPI_RETURNの戻り値にエラー状態を示す値が設定されなければならない。
・より良くサポートされた、最も近い値がMaxFMRRequestedの値として用意されなければならない；しかし、BSPはサポートされている値（FMRAchieved）を戻さなければならない。（BSPはコンポーネントレジストリー内に粗いスコア値の使用機能が装備されているかどうかを提示しなければならない。）
注記）閾値とスコア化に関して、FMRの使用についての説明がある本規格書C．４節を参照のこと。
・Payloadの戻り値は、もしこれが入力ReferenceTemplateと連携され、スコアがFMRAchievedの値（BSPはコンポーネント・レジストリー内に、ペイロードが戻される場合に必要な最低FMRを挿入しておかなければならない）より良い場合にのみ必要である。
・AdaptedBIRの戻り値はオプションである。

5) BioAPI準拠識別エンジン

・BioAPI準拠識別エンジンの要求仕様は、次の例外を除き照合エンジンと同じである。すなわち、BioSPI_CreateTemplate関数は、Purposeの値、
BioAPI_PURPOSE_ENROLL_FOR_IDENTIFICATION_ONLYと共にもまたCapturedBIRを受け取り、BioSPI_Process関数は、以下の追加関数と同様にPurposeの値、BioAPI_PURPOSE_IDENTIFYと共にもまたCapturedBIRを受け取らなければならない。

- BioSPI_IdentifyMatch

(a) BioSPI_IdentifyMatch

・より良くサポートされた、もっとも近いMaxFMRRequestedの値のみが与えられなければならない。一致した候補の戻り値が必要である；しかし、BSPは、次に近い値/増加分をFMRAchievedのフィールドに対する戻り値として戻すことができる。（BSPはコンポーネント・レジストリー内に粗いスコアー処理を装備しているかどうかを提示しなければならない。）
注記）本規格書C．４節のスコアー処理と閾値処理に関する説明を参照すること。
・binningのサポートはオプションである。

6) オプション機能

・次の機能はBSPのサポートと準拠に関して、オプションと見なされる。次のことに注意するこ

と。

- 実装される場合は、オプション機能は定義された仕様に適合しなければならない。
- BSPはコンポーネント・レジストリー内に（BioAPI_Util_InstallBSPレジストリ関数内のBSPSchemaの要素OptionsMaskにより）各オプション機能がサポートされているかどうかを（インストール時に）登録しなければならない。
- BSPの仕様定義文章内には、どのオプションがサポートされているかおよびサポートされていないかを判定するための表が含まれなければならない。

(a) オプション関数

① 基本関数

・オプションではあるが、基礎技術としてサポートされる場合、次の基本関数が全てのBSPにおいてサポートされることを強く推奨する。

ア） BioSPI_Capture

・もしこの関数がサポートされる場合は、照合BSPはPurposeの値として、BioAPI_PURPOSE_VERIFYまたはBioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLYのみを受理できることが必要である。もし、サポートされていない他の目的の値が定義された場合は、BioAPI_RETURNとしてエラー値が設定されなければならない。この関数は目的の値として、それぞれBioAPI_VERIFY、BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLYまたはBioAPI_PURPOSE_ENROLLと共にCapturedBIRを戻す。
・もし、この関数がサポートされるならば、たとえ戻り値のデータの内容およびフォーマットについて差異が全くない場合でも、識別BSPは、全ての目的の値（例外として、BioAPI_PURPOSE_AUDIT）を受理しなければならない。
・AcceptanceとSubtypeの使用はオプションである。
・もしBSPが複数のBIRデータフォーマット出力をサポートする場合は（コンポーネント・レジストリー内のBSPスキーマに指定されているように）、OutputFormat入力とBIR中にProcessedBIRの戻り値が受け取られなければならない。
・生データ（AuditData）の戻り値はオプションである。
・デフォルトとして、BSPはBioSPI_Captureの動作に連携して何らかのGUIを用意しなければならない。しかし、アプリケーションによるこのGUIの制御をサポートすることはオプションである。

（イ） BioSPI_CreateTemplate

・もしこの関数がサポートされる場合は、BSPは、Purpose値、BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLYと共にのみ入力CapturedBIRを受けとらねばならない。もし、他のサポートされない目的の値が定義された場合は、BioAPI_RETURNの値としてエラーが設定されなければならない。
・もしこの関数がサポートされるならば、識別BSPは全てのEnroll（登録）の目的の値について入

力CapturedBIRを受け取らなければならない。
・Payloadの受け取りはオプションである。テンプレートの更新（ReferenceTemplateを通して）は、オプションである。
・もしBSPが複数のBIRデータフォーマット出力をサポートする場合は（コンポーネント・レジストリー内のBSPスキーマに指定されているように）、OutputFormat入力とBIR中にProcessedBIRの戻り値が受け取られなければならない。

（ウ）　BioSPI_Process
・もし、この関数がサポートされる場合は、照合BSPはPurpose値、BioAPI_PURPOSE_VERIFYと共にのみ入力CapturedBIRを受けとらねばならない。もし、他のサポートされない目的の値が定義された場合は、BioAPI_RETURNの値としてエラーが設定されなければならない。
・もしこの関数がサポートされるならば、識別BSPは、たとえ戻り値のデータの内容およびフォーマットについて差異が全くない場合でもPurpose値、BioAPI_PURPOSE_VERIFYまたはBioAPI_PURPOSE_IDENTIFYと共にのみ入力CapturedBIRを受け取らなければならない。
・もしBSPが複数のBIRデータフォーマット出力をサポートする場合は（コンポーネント・レジストリー内のBSPスキーマに指定されているように）、OutputFormat入力とBIR中にProcessedBIRの戻り値が受け取られなければならない。

（エ）　BioSPI_ProcessWithAuxBIR
・もし、この関数がサポートされる場合は、照合BSPはPurpose値、BioAPI_PURPOSE_VERIFYと共にのみ入力CapturedBIRを受けとらねばならない。もし、他のサポートされない目的の値が定義された場合は、BioAPI_RETURNの値としてエラーが設定されなければならない。
・もしこの関数がサポートされるならば、識別BSPは、たとえ戻り値のデータの内容およびフォーマットについて差異が全くない場合でもPurpose値、BioAPI_PURPOSE_VERIFYまたはBioAPI_PURPOSE_IDENTIFYと共にのみ入力CapturedBIRを受け取らなければならない。
・BSPがこの関数をサポートする場合、この関数が入力として受け取れるAuxiliaryDataの内容とフォーマット（または、発生源について）の記述が含まれるであろう。
・もしBSPが複数のBIRデータフォーマット出力をサポートする場合は（コンポーネント・レジストリー内のBSPスキーマに指定されているように）、OutputFormat入力とBIR中にProcessedBIRの戻り値が受け取られなければならない。

（オ）　BioSPI_VerifyMatch
・もし、この関数がサポートされる場合は、Purpose値、BioAPI_PURPOSE_VERIFYと共にのみ入力BIR（ProcessedBIR）を、またPurpose値、BioAPI_PURPOSE_ENROLLまたはBioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLYと共にReferenceTemplateを受けとらねばならない。もし、他のサポートされない目的の値が定義された場合は、BioAPI_RETURNの値としてエラーが設定されなければならない。
・より良くサポートされた、もっとも近いMaxFMRRequestedの値のみが与えられなければならない。

一致した候補の戻り値が必要である；しかし、BSPは、次に近い値/増加分をFMRAchievedのフィールドに対する戻り値として戻すことができる。BSPはコンポーネント・レジストリー内に粗いスコアー処理を装備しているかどうかを提示しなければならない。）
注記）本規格書C．４節のスコアー処理と閾値処理に関する説明を参照すること。
・Payloadの戻り値は、もしこれが入力eferenceTemplateと連携され、スコアがFMRAchievedの値より良い場合にのみ必要である。（BSPはコンポーネント・レジストリー内に、ペイロードが戻される場合に必要な最低FMRを挿入しておかなければならない）
・AdaptedBIRの戻り値はオプションである。

（カ） BioSPI_IdentifyMatch
・もし、この関数がサポートされる場合は、Purpose値、BioAPI_PURPOSE_IDENTIFYと共にのみ入力BIR（ProcessedBIR）を受けとらねばならない。もし、他のサポートされない目的の値が定義された場合は、BioAPI_RETURNの値としてエラーが設定されなければならない。
・より良くサポートされた、もっとも近いMaxFMRRequestedの値のみが与えられなければならない。
一致した候補の戻り値が必要である；しかし、BSPは、次に近い値/増加分をFMRAchievedのフィールドに対する戻り値として戻すことができる。（BSPはコンポーネント・レジストリー内に粗いスコアー処理を装備しているかどうかを提示しなければならない。）
注記）本規格書C．４節のスコアー処理と閾値処理に関する説明を参照すること。
・binningのサポートはオプションである。

② データベース操作
・BSPは、内部に（またはBSP制御の）BIRデータベースを用意することを必須とはしていない。しかし、もし用意されるならば、データベースへのアクセスとメンテナンス管理のために全てのデータベース関数が用意されなければならない。用意された全てのデータベース関数は、その定義に適合しなければならない。

③ BioSPI_Import
・この関数は、全体としてオプションである。もし用意されるならば、コンポーネント・レジストリに反映されなければならない。そして定義に従い実装されなければならない。
・もし、目的（Purpose）の値がBioAPI_PURPOSE_ENROLLまたは
BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLYならば、インポートされたデータの処理のためには、照合BSPのみが必要である。識別BSPは、目的（Purpose）の値が任意のEnroll（登録）目的値である場合のみ、インポートされたデータの処理のために必要である。
・もしBSPが複数のBIRデータフォーマット出力をサポートする場合は（コンポーネント・レジストリー内のBSPスキーマに指定されているように）、OutputFormat入力とBIR中にConstructedBIRの戻り値が受け取られなければならない。

④　BioSPI_ Preset Identify Population

・この関数は、全体としてオプションである。もし用意されるならば、コンポーネント・レジストリに反映されなければならない。そして定義に従い実装されなければならない。

⑤　SPI BioAPIユニット操作

・全てのBioAPIユニットの操作は、全体としてオプションである。もし用意されるならば、コンポーネント・レジストリに反映されなければならない。そして定義に従い実装されなければならない。

（b）　オプションの付属関数

・次の表はオプション機能の確認のために記載されており、各項目はBSPがサポートしているか、またはしていないか（BSPの記述文書およびコンポーネント・レジストリ内のBSPスキーマにおけるオプションのマスクで）を公表しなければならない。

表３－２ -オプション付属機能

機能	サポート有	サポート無し
生(検査)データの戻り値		
品質値の戻り値		
GUIのアプリケーション制御		
GUIの流れに対するコールバック		
測定源の存在検出		
ペイロード		
BIRの署名		
BIRの暗号化		
テンプレートの更新		
アダプテーション		
binning		
自己完結型装置		
カード内照合		
キャプチャに対するSubtype		
センサーBFP		
アーカイブBFP		
照合BFP		
処理BFP		
粗いスコア付け		

①　生データの戻り値

・センサーからの生体認証データの収集を含む機能が、表示や検査の目的で生データの戻り値をオプションとしてサポートできる。この機能がサポートされている場合、出力パラメータAuditDataがこのデータへのポインター値を含んでしる。サポートされない場合は、BioAPI_UNSUPPORTED_BIR_HANDLE値がBSPから戻される。

② 品質値の戻り値

・センサーからの新しい生体認証データの収集において、BSPは、このデータに関連する相対的な品質値を算出することが出来る。この値は、CapturedBIR（及び、オプションのAuditData）の戻り値のヘッダー部に含まれる。もしサポートされている場合は、このヘッダーのフィールド内に１から１００の間の正の値が挿入されている。サポートされない場合は、このフィールドに－２の値が入れられている。これらは、BioSPI_CaptureおよびBioSPI_Enroll関数実行時に発生する。

・同様に、BIRが処理される過程において、別の品質値の演算処理を実行でき、この品質値はProcessedBIR（およびオプションのAuditData）のヘッダー内に含ませられる。これは、BioSPI_CreateTemplate、BioSPI_Process、BioSPI_ProcessWithAuxBIR、BioSPI_Verify、BioSPI_VerifyMatch、BioSPI_Enroll、およびBioSPI_Import（ConstructedBIR）関数の実行中に発生する。

・BSPは、コンポーネント・レジストリ（OptionsMask）に、各タイプのBIR－生データ、中間、処理済みーそれぞれに対して品質測定値の演算処理がサポートされているかどうかを書き込まなければならない。

注記）品質データの使用に関する追加情報についは、本規格書７．４９節を参照すること

③ アプリケーションによるGUIの制御

・BSPはアプリケーションにGUIの画面表示の変化（Look and Feel）の制御を許容するために必要なコールバックをサポートする。

・全てのBioAPI準拠のBSPは、デフォルトとして、キャプチャ操作の過程においてユーザインターフェース情報を表示する機能を用意しなければならない。（そのような表示ディスプレイが存在し、生体認証技術に必要であると仮定すると）全てのBSPが供給するGUIには、オペレータが中止/キャンセルを実行できる機能が含まれていなければならない。

・オプションとして、BSPはまたアプリケーションが制御するGUIをサポートすることが出来る。この場合、BSPはBioSPI_SetGUICallbacks関数をサポートしなければならない。

・もしアプリケーション制御のGUIがサポートされる場合、BSPは追加として、データの流れ（StreamingData）を用意することが出来る。（例えば、音声や顔認証）もし、データの流れが用意できた場合は、BSPは入力パラメータ、GuiStreamingCallbackおよびGuiStreamingCallbackCtxをサポートしなければならない。

・すべてのBSPは、それがサポートする全てのGUI機能/オプションをコンポーネント・レジストリに書き込まなければならない。

注記）ユーザーインターフェースに対する考察に関する追加情報は本規格書Ｃ７節を参照すること

④　GUIストリーム・コールバック

・BSPはGUIストリームデータを用意し、GUIストリームコールバックをサポートする。

⑤　測定源の存在検出

・BSPはサンプルの収集が可能であるタイミングを検出できる。そして測定源存在イベントをサポートする。

注記）「サンプル収集可能」の一つの例として、指紋センサーのプラテンと指が物理的に接触したタイミングがそうであろう。

⑥　ペイロードの搬送

・いくつかのBSPは、基準となるテンプレートの生成時における「ペイロード」の保存、および後の照合成功時におけるそのペイロードの開放をオプションとしてサポートできる。ペイロードの内容には制限が無い。しかし、PINやBIRに生体情報が取り込まれている利用者に関連するプライベート鍵のような秘密鍵を含むことができる。ペイロードのデータはBioSPI_Enroll、またはBioSPI_CreateTemplate関数実行中に用意され、BioSPI_VerifyまたはBioSPI_VerifyMatch関数の実行結果として開放される。サポートされる場合は、BSPはコンポーネント・レジストリ内に収容可能な最大サイズを書き込まなければならない。最大サイズ　’０’はペイロードの搬送がサポートされていないことを意味する。入力されたペイロードの大きさがこの値を超えた場合はエ

ラー発生として処理される。ペイロード搬送がサポートされない場合は、出力Payloadは’－１’に設定される。

注記）ペイロードに関する追加情報は本規格書Ｃ．５を参照のこと

⑦　セキュリティ機能

・BioAPIは、必須ではないが、セキュリティ機能をサポートする。追加として、BioAPIの境界を横断する機能は定義しない。

・BSPは次の機能を提供しているかどうかをコンポーネント・レジストリに書き込まなければならない。

- BIRの暗号化
- BIRの署名

注記）この機能に関連する鍵管理については、本APIでは取り扱わない。

⑧　テンプレートの更新

・CreateTemplateおよびEnroll関数が以前登録したテンプレート（ReferenceTemplate）の更新（すなわち、BSPはこのテンプレートを新しくキャプチャしたサンプルと共にNewTemplateを生成するために）のためにアプリケーションによって用意される。BSPは全ての場合において、NewTemplateを戻さなければならない。（エラー発生の場合を除いて）しかし、テンプレートの更新がサポートされていない場合は、用意されたReferenceTemplateをその計算処理において無

視することができる。

⑨　テンプレートの採用

・いくつかのBSPはオプションとして新しくキャプチャした生体認証データを基準のBIRの更新に使用する機能を提供できる。これはBioSPI_VerifyまたはBioSPI_VerifyMatchが成功裏に終わった場合（すなわち、Result = BioAPI_TRUE）にのみ実行可能である。これは、登録されたBIRを可能な限り最も高品質な状態で、出来るだけ新鮮に保つために実行される。BSPはいつおよび採用していいかどうかの決定を実施する。（品質、経過時間、及び差異の大きさ等の要因を基として）もし、BSPが採用をサポートしていない場合は、AdaptedBIRの戻り値が’－２’（BioAPI_UNSUPPORTED_BIR_HANDLE）に設定される。もしBSPが採用をサポートし、しかし。何らかの理由により使用できないか、または採用の不実行を選択する場合は、AdaptedBIRの戻り値が’－１’（BioAPI_INVALID_BIR_HANDLE）に設定される。BSPは採用に対するサポートをコンポーネント・レジストリ（OptionsMask）に書き込まなければならい。

⑩　Binning

・識別BSPはオプションとして、応答時間の改善のため、探索するデータベースの集団数を制限する方法をサポートすることができる。

・これは識別型のBSPにのみ応用でき、BioSPI_IdentifyおよびBioSPI_IdentifyMatch関数の操作中にのみ発生する。識別BSPはbinningのサポート、否サポートをコンポーネント・レジストリに書き込まなければならない。binningをサポートしない識別BSPは、入力BinningのON/OFFパラメータを無視できる。

注記）BinningのON/OFF以外のBinningに対する方針の変更や設定に関する明確なサポートは用意されていない。

⑪　自己完結型デバイス

・サポートされるセンサーデバイスは「自己完結型」である。すなわち、センサー自身以外に、少なくとも照合機能がデバイス内に完全に含まれ、そしてそのデバイスは１つのBIRデータベースもまた含むことができる。

注記）自己完結型デバイスは２つ以上のBioAPIユニットを含むことが出来る。（すなわち、センサーユニットに加えて他の分類に属する１つのBioAPIユニット）

⑫　カード内照合（マッチオンカード）

・BSPは、スマートカード上で実行される照合操作をサポートする機能を有する。BSPのサブクラスに依存するが、これはBSPがMOCに適したBIRの生成や操作を実行できることを要求している。さらに、BSPはカードとの相互作用（間接および直接的に）が可能でを要求する。

⑬　サブタイプのキャプチャ

・BSPは、アプリケーションが、与えらたタイプの中のどの生体認証サブタイプに対してキャプ

チャすること（例えば、右人指し指または左目）が出来る機能をサポートしている。もしこのオプションがサポートされ、そしてアプリケーションがキャプチャするサブタイプを指定するならば、BSPはその指定された生体認証の特徴値をキャプチャし、戻り値CapturedBIRのヘッダー内に示す。これは生体認証のキャプチャが必要な全ての関数に当てはまる。（すなわち、BioSPI_Capture、BioSPI_Enroll、BioSPI_VerifyまたはBioSPI_Identifyである）。

⑭　センサーBFP

・BSPは連携したFPIを通して様々なセンサーデバイス（ユニット）をサポートする。装着されたときにBSPは、生体認証のキャプチャに必要な任意の関数（すなわち、BioSPI_Capture、BioSPI_Enroll、BioSPI_VerifyまたはBioSPI_Identify）に対してセンサーユニットを使用する。

⑮　アーカイブBFP

・BSPは任意の内部アーカイブ機能に加えて、様々なアーカイブユニットの装着をサポートする。BSPはこのように連携するFPIを通してそのようなユニットとのインターフェースをサポートする。これはまたBSPがBioSPIデータベース操作をサポートすることを必要としている。装着されると、BSPは、このアーカイブに連携するDatabase UUIDが任意のデータベース関数のために指定されたときに、このアーカイブを使用する。

⑯　照合BFP

・BSPは交換可能な照合ユニット（アルゴリズム）の装着をサポートする。装着されたとき、BSPはその交換可能なユニットをそのユニットが可能な任意の照合操作のために使用する。（すなわち、もしそれが１対照合モジュールならば、照合操作のために）

⑰　加工処理BFP

・BSPは交換可能な加工ユニット（アルゴリズム）の装着をサポートする。装着時、BSPはその交換ユニットを任意のBioSPI_ProcessまたはBioSPI_CreateTemplate関数呼び出しに利用する。

⑱　粗いスコア処理

- BSPは、加算されたスコアー値の戻り値をサポートする。これは、特別な攻撃シナリオ（本規格書Ｃ．４．６章参照）に対処することを目的として加算処理されている。このオプションがOptionsMaskに示されている場合以外は、BSPは微細に量子化されたスコアを提供する。

３．２　BioAPI Liteについて

３．２．１　はじめに

2006年に公布されたISO/IEC19784-1BioAPI規格に対して、フレームワークを使用せず、使用関数の数を制限したISO/IEC19784-3: Information Technology – Biometric Application programming Interface – Part 3: BioAPI Liteという名称で2005年3月に英国よりNP提案があった。このNP提案の中で、BioAPI Liteの定義と適用範囲および３つの分類について議論し詳細を決定することが記載されていた。すなわち、BioAPI Liteの定義として、BioAPIの縮小形として生体認証センサーを含む組み込みシステムに適用できる仕様を策定する。

適用範囲としては

（ａ）BioAPIフレームワークは使用せず、１つのアプリケーションとBSPが直接相互にコミュニケートするシステム

（ｂ）最小のBioAPIフレームワークと１つのアプリケーションおよび複数のBSPから構成されるシステム

（ｃ）最小のBioAPIフレームワークと複数のアプリケーションおよびBSPから構成されるシステム

が含まれるが、その具体的対象製品、および適用範囲に関してその後、議論がISO/IEC JTC1/SC37 WG2小委員会においてSpecial Groupにより続けられてきたが、合意が得られず昨年、期限切れのため規格開発活動は中止となった。しかし、2007年１月のイスラエル会議において、再度、スペインがNP提案することとなった。ここで、NP提案と共にSC37/WG2委員への回覧が開始されたベースドキュメントの内容をまとめて紹介する。

３．２．２　New Work Item Proposal(NWI)の内容

本提案は再提案であり、WDの提案も同時に行っている。BioAPI　Liteの説明としてCPU性能とメモリー容量が制限された組み込みシステムに対する標準規格を意図することが記載されている。この組み込みシステムの設計においては、生体認証技術の詳細については適用範囲外と定義している。ＯＳや高級言語を利用するソフトウェアを使用しないバイオメトリックモジュールに対する相互交信のインターフェイス定義であるが、より低レベルのハードウェアに関連した実装方式はスコープ外としている。

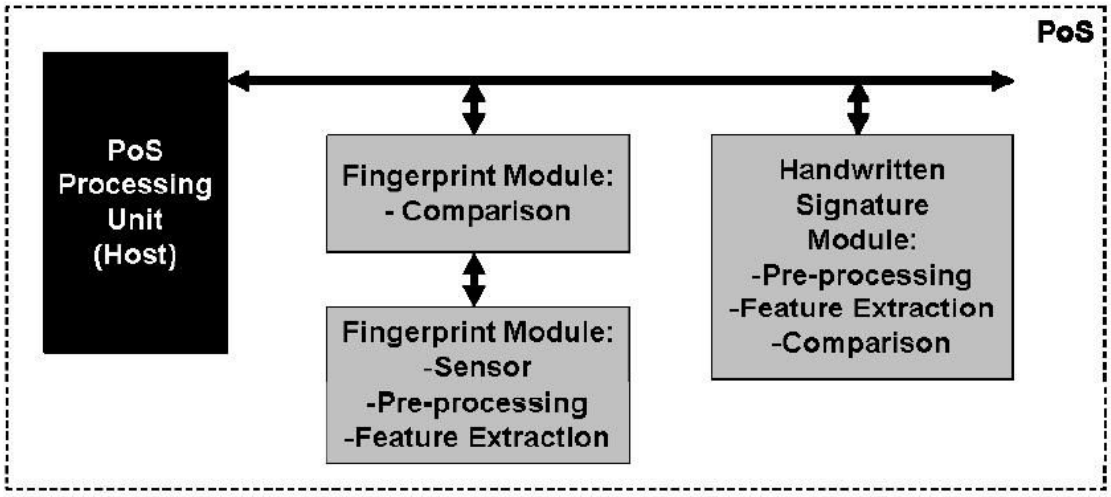
３．２．３　ベースドキュメントについて

ここでは、SC37に回覧されているN2453 BioAPI Lite WDについてＷ記載する。

適用範囲として、以前のNP提案と同じように、ＣＰＵ性能およびメモリー容量が制限されている組み込みシステムに統合されたバイオメトリック・モジュールに対する標準インターフェイスを定義している。また、ＯＳや高級言語を使用せず、バイオメトリック・モジュールと直結した部分のインターフェイスをコマンドのメッセージ文のフォーマットとして定義している。ハードウェアに関連した低レベルの実装形態は本規格の適用外としている。

BioAPI Liteが用いられる環境として、組み込みシステムが上げられ、典型的な例として（お店でのクレジットカードによる支払いに使用される）OEMモジュールを利用したPoS端末を上げてい

る。このPoS端末では、サインと指紋認証が使用されている場合が次図に例示されている。



OEMモジュールに基づくバイオメトリック・システムの例

この例では、ＩＣ化されたハードウェアモジュールがサイン認証に必要な機能（センサー、バイオメトリック生サンプルデータ（ISO/IEC 19704-7）、および利用者のテンプレート）を提供する。また、指紋認証については、２つのモジュールから構成され、センサーモジュールと照合を行うコンパリゾン・モジュールから構成される。

この例より、BioAPI Liteは、フル装備の完全なBioAPIとは次の点で異なると定義している。

a）BioAPI フレームワークとコんポーネント・レジストリは使用しない。（すなわち、アプリケーションとBSPは直結されている。）

b）BioAPI機能の一部分を使用する

c）状態表示機能はない

d）データハンドルは使用しない

e）ファンクション・プロバイダー・インターフェイス(FPI)はサポートしない。

f）オプション機能は最小とする。

しかし、生体認証機能については、登録、照合機能、および　データ形式（バイオメトリック・インフォメーション・レコード(BIR)）については、BioAPIの定義に従うことが記載されている。

BioAPI LiteのBSPコンポーネントは、一般的に特別なセンサーデバイスとファームウェア、およびソフトウェアから構成されると定義している。

典型的な組み込みシステムは次の様な環境におかれている。

a)　メモリー容量の制限

b)　プロセッサーの大きさ、速度の制限

c)　制限されたＯＳの使用

d)　１つのハードウェア結合されたセンサー

e) スタンドアロン（ネットワーク接続されていない）

例として、リモコン装置、倉庫開閉装置、自動車の自動イグニション装置、入退出装置、メモリースティック、各種認証トークン装置、および携帯型武器が上げられている。
これらに共通なユーティリティの標準インターフェイスとして、次の項目が可能となるように定義する。

i) 製造者が異なるセンサー（指紋と顔のセンサー）を使用する場合でも、同じプログラムのコードを基本として使用できるようにする。

ii) OEM センサー製造者が、複数のデバイスベンダーに、単一のファームウェアで異なる種類のセンサーを組み込み供給できるようにする。

・BioAPI　Liteの一般的な構造
２つの構成要素から成り、その内の１つはホスト（一般処理ユニット）と呼ばれる装置全体の機能を管理し、バイオメトリック機能（照合、登録、データ処理および保存）を実行するOEMユニットに直結されている。BioAPI Liteは組み込み型バイオメトリックOEMモジュールとの統合をサポートする。この統合のためにサポートされるオプションとして

①タイプ１　―　ＩＣ化されたOEMモジュール：１つのOEMモジュール内にバイオメトリックセンサー、メモリ、アルゴリズムが含まれる

②タイプ２　―　組み合わせ型OEMモジュール：１つのOEMモジュールが次の１つまたは複数の機能を含む

i) バイオメトリック・センサー

ii) バイオメトリック・メモリー

iii) バイオメトリック・アルゴリズム

タイプ１の例として、リモートコントロール用OEM指紋センサーで、指紋センサー、データ処理、ファームウェア（照合アルゴリズム、バイオメトリックデータ処理を含む）および５指のテンプレートを保存できる回路基板内メモリーを含む

タイプ２の実装例は、顔写真を取る一般コンシューマー製品としてのCCDカメラであり、顔認証を行う。この場合、基準テンプレートの保存が発生し、これはアプリケーションソフトウェアにより独立に制御されるかまたはバイオメトリック・モジュールにより制御されるとしている。

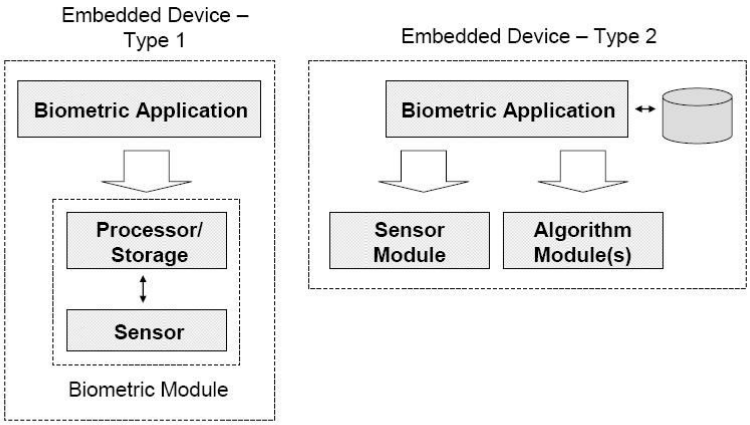


Figure 2 — Different ways to implement OEM biometric modules

図３－６　タイプ１およびタイプ２の実装例

この２つのタイプは次の図のように一般化されている。この図で、BioAPI Liteに適合するモジュールは、ホストシステムが要求するバイオメトリック機能の全てまたは一部を実行し、BioAPIにより標準インターフェイスを通してホストシステムと交信する。このインターフェイスをBioAPI Liteと呼ぶことにすると記載されている。オプションとしてこのモジュールは他のBioAPI Liteモジュールとも交信することができるようになっている。

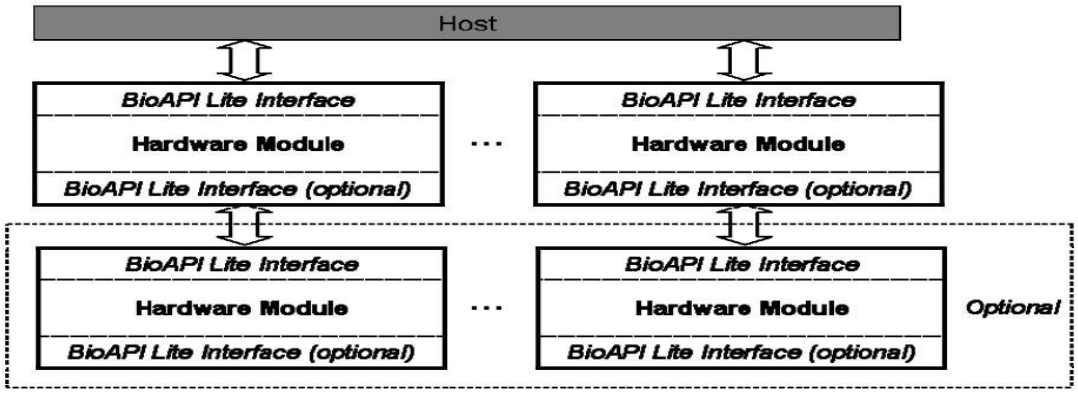


Figure 3 — General Architecture of a BioAPI Lite system

図３－７　BioAPI Liteシステムのアーキテクチャー

・フレームの構造

BioAPI Liteモジュールとホストの交信メッセージを考察する場合の制限事項として、

1)モジュール間での特徴ベクトルまたはテンプレート・データ転送は65KB程度であり、生データは更に大きくなる。

2)データ長は固定ではないので、伝送はフレーム分割され、各フレームは可変長で2バイトのフィールド長を持つ。

3)フレーム構造と、フレーム操作のためのプロトコルが必要である。

4)処理時間はアプリケーションに依存するが、タイムアウト処理は最小100msから最大数秒の間でなければならない。

5)非同期交信が推奨されるが、同期、非同期は実際の実装方式に依存する。

6)セキュリティ機能が必要である。

7)データ転送の周波数帯域が保証されなければならない。モジュール間の伝送において１フレーム/100msが最低必要であり、これは最低5Mbpsに当たる。

これらの要求事項を基に次の定義が下されている。

・フレームの定義

BioAPI Lite使用環境においては、１つのマスタ/複数のスレーブ間のプロトコルが定義される。マスターはホストであり、スレーブは、様々なバイオメトリック・モジュールである。

カスケード形のモジュール配置では、１つのバイオメトリック・モジュールが他のモジュールと連携し、BioAPIを実装しマスターモードでの他のモジュールと交信する。

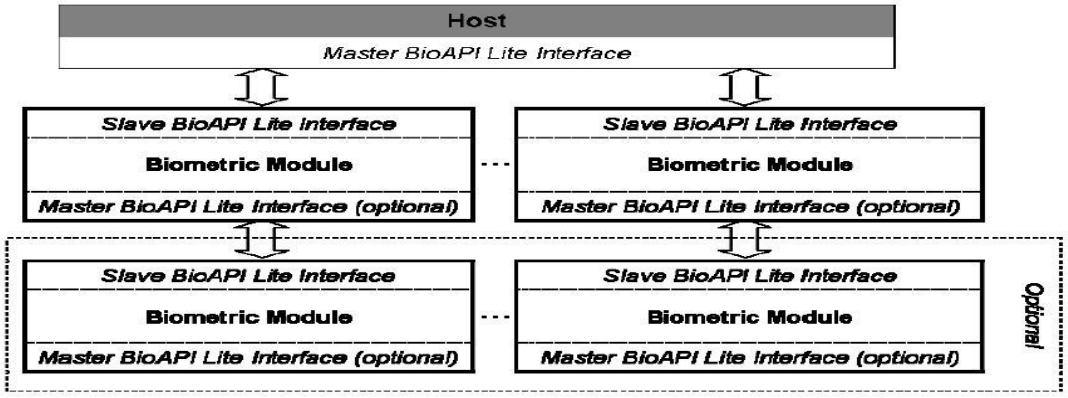


Figure 4 — Master / Slave configuration of BioAPI Interface

図３－８　BioAPIインタフェースのマスター／スレーブシステム

複数のモジュールがホストに接続されるので、フレームはスレーブのアドレスを保持しなければならない。１つのアプリケーションにおいて複数のモジュールを考慮する必要があるので、このモジュール数は制限されるが、実際のアドレスは考慮せず、識別フィールド(Slave IDフィールド)を定義している。

命令文とそれへの応答が定義されている。伝送データの総数は、最初に2^{16}バイトに制限されデータ長フィールドは２バイトが必要である。

単一マスター／スレーブシステムについて設計されると、命令／応答の組み合わせが定義される。命令は常にホストから送信され、応答は常にスレーブから発信される。コマンドの数は256以下であり、命令識別には１バイトのフィールドが必要である。

完全性チェックのためのフィールドが必須である。リプレイアタック対処のため、セッションキーの使用を薦めており、このために2バイトのフィールドが確保されている。

エラー処理は使用する低レベルのプロトコルの規定による。

2^{16}バイト以上のデータを交換するためにフレームを結合する場合、最上位ビットである[S]フィールドが使用され、1の場合は、続きのフレームが存在し0の場合はそれ以上の送信フレームが存在しないことを表す。フレームが結合される場合、それに続く全てのフレームでは、結合されたフレームの１番目に関連した２番目のバイト（CommandまたはStatus/Error）を元のままに保持しなければならない。また、[S]フィールドの7つの下位ビットも同様である。

表３－３、表３－４　フレームの構造

Table 1 — Structure of the Command Frame

Slave ID [S] (1 byte)	Command [C] (1 byte)	Data Length [L] (2 bytes)	Data [D] (L bytes)	Integrity Check (2 bytes)

Table 2 — Structure of the Response Frame

Slave ID [S] (1 byte)	Status/Error [E] (1 byte)	Data Length [L] (2 bytes)	Data [D] (L bytes)	Integrity Check (2 bytes)

・コマンドの定義

BioAPI Liteに含まれるコマンドについて次の項目を用いて解説している。

・①　コマンドの説明

・②　コマンド内で送信される入力パラメータ

・③　バイオメトリック・モジュールが送信する応答文

・④　フレーム構造（１６進値、可変シンボル）

・⑤　SDK 開発用 C 関数のヘッダー（戻り値、ステータス/エラーコード）

コマンドは次のように分類されている。

１）管理コマンド

a)モジュールの初期化（必須）

b)モジュールのクローズ（必須）

c)GetProperties（必須）

d)UpdateParameters

２）登録関連コマンド

a)Capture4Enrol

b)Acquire4Enrol

c)Enrol

d)EraseEnrolments（テンプレート保存機能を持つデバイスの場合は必須）

３）バイオメトリック処理コマンド

a)Capture

b)CaptueProcess

c)CaptureCompareExternal

d)CaptureCompareInternal

e)Process

f)ProcessCompareExternal

g)ProcessCompareInternal

h)CompareExternal

i)CompareInternal

４）テンプレート管理コマンド

a)StoreTemplate（ボード内テンプレート保存）

b)RetrieveTemplate（ボード内テンプレート保存）

・データのタイプ、形式およびコード表示

１）スレーブIDフィールド[S]

表３－５　、スレーブIDの[S]フィールドコーディング

Table 3 — Slave ID field [S] coding

b7	b6	b5	b4	b3	b2	b1	b0	
0								No further frame to be sent
1								Further frame is to be sent
	x	x	x	x	x			Modality in 5 bits (following coding in Table 6 — Modality / Data format coding)
						0	0	Capture functionality module
						0	1	Pre-processing functionality
						1	0	Feature extraction functionality
						1	1	Comparison functionality

注記　BioAPI Liteの動作環境として、バイオメトリクスのモードと機能については１モジュールの使用が想定されている。

１つのモジュールが複数の機能を持つ場合、最大数が使用される。すなわち、ICモジュールを考える場合は、[S]フィールドは11で終わっていなければならない。

・表３－６コマンド・フィールド[C]

Table 4 — Command field [C] coding

b7	b6	b5	b4	b3	b2	b1	b0	Hex value	
0	0								**Management Commands**
0	0	0	1	1	0	0	1	19	Initialize
0	0	0	0	0	1	1	0	06	Close
0	0	0	0	0	0	0	1	01	Get Properties
0	0	0	0	0	0	1	0	02	UpdateParameters
0	0	x	x	x	x	x	x	xx	RFU
0	1								**Template Management Commands**
0	1	0	0	0	0	0	1	41	Store Template
0	1	0	0	0	0	1	0	42	Retrieve Template
0	1	x	x	x	x	x	x	xx	RFU
1	0								**Enrolment Commands**
1	0	0	0	0	0	0	1	81	Capture4Enrol
1	0	0	0	0	0	1	0	82	Acquire4Enrol
1	0	0	0	1	0	0	0	88	Enrol
1	0	0	1	0	1	1	0	96	Erase Enrolments
1	0	x	x	x	x	x	x	xx	RFU
1	1								**Biometric Process Commands**
1	1	0	0	0	0	0	1	C1	Capture
1	1	0	0	0	0	1	0	C2	Process
1	1	0	0	0	0	1	1	C3	CaptureProcess
1	1	0	0	0	1	0	0	C4	CompareExternal
1	1	0	0	0	1	1	0	C6	ProcessCompareExternal
1	1	0	0	0	1	1	1	C7	CaptureCompareExternal
1	1	0	0	1	1	0	0	CC	CompareInternal
1	1	0	0	1	1	1	0	CE	ProcessCompareInternal
1	1	0	0	1	1	1	1	CF	CaptureCompareInternal
1	1	x	x	x	x	x	x	xx	RFU

・表３－７ステータス/エラーフィールド[E]

Table 5 — Status/Error field [E] coding

b7	b6	b5	b4	b3	b2	b1	b0	C Definition	
0									**Command processed correctly**
0	0	0	0	0	0	0	0	NO_ERROR	No error
0	0	0	0	0	0	1	0	MATCH	Match
0	1	1	1	1	1	0	1	NO_MATCH	No Match
0	0	0	0	0	1	0	1	FTA	Failed to Acquire (FTA)
0	0	0	0	1	0	0	1	FTE	Failed to Enrol (FTE)
0	0	0	1	0	0	0	1	NO_QUALITY	Quality requirements not accomplished
0	0	1	0	0	0	1	1	WAIT	Wait, still processing
0	x	x	x	x	x	x	x		RFU
1									**Error in command processing**
1	0	0	1	1	0	0	1	TIME_OUT	Time-out
1	0	0	1	1	0	1	0	UNK_COMMAND	Command not supported
1	0	0	1	1	1	0	0	UNDEFINED	Unexpected/undefined
1	x	x	x	x	x	x	x		RFU

・バイオメトリックモダリティのコード化

ISO/IEC JTC1/SC37 Standing Document 9に従い、バイオメトリックモダリティのコードを定めている。

３．３　ISO／IEC24709－1:2007Information technolog－Conformance testing for the biometric application programming interface(BioAPI)－Part1:Methods and procedures について

３．３．１　はじめに

ISO/IEC 24709は、BioAPI 適合性評価試験 という代表題目のもと、4つのパートから構成されている。

Part1 ： 方法と手続

Part2 ： バイオメトリック サービス プロバイダー に対する試験

Part3 ： BioAPI フレームワークに対する試験

Part4 ： バイオメトリック アプリケーションに対する試験

本書では、24709-1の要約を纏める。

３．３．２　24709-1の文書構成

本規格は以下の章から構成される。

緒言/ はじめに

第1章は目的

第2章は適合性評価の意味

第3章は基準となる見本

第4章は用語と定義

第5章は略語

第6章はBioAPIに対する一般的なテスト方法と適合性評価試験モデルについて説明する。

第7章は適合性評価試験を表明する定義に使われたXMLに基づく主張言語を定義する。

第8章は主張言語の要素を定義する。

第9章は適合性評価テストにおける、BioAPIの標準のBioAPIインタフェース機能の使用を指定する。

第10章は主張言語に組み込まれている変数を定義する。

第11章は、XML構文を使用した試験履歴記録を定義する。

第12章は、XML構文を使用した試験報告書を定義する。

第13章はBioAPI適合性評価試験ソフトウエアの一般的な概念と構造について説明する。

付録Aは、標準を定めており、表意言語のXML図式を定義する。

付録Bは、標準を定めており、表意言語のASN.1図式を定義する。

付録Cは、標準を定めており、適合性評価試験履歴記録のためにXML図式を定義する。

付録Dは、参考になるもので、適合性評価試験の手順という要素と試験のカテゴリーの表記を含んでいるBioAPIテスト方法実現の入門書について説明する。

付録Eは、参考になるもので、BioAPIの適合性評価試験の手順の全体に対する一般的なフレームワークを説明する。

付録Fは、参考になるもので、自己主張言語に関する関係図を提供する。

参考文献一覧は、多くの標準化組織を参照しています。ISO・IEC・NIST・IEEEを含んでおり、一般的に適合性評価評価に関係した、特に適合性評価試験に関係した多くの文書やホワイトペーパー出版した他の組織を含んでいる。

３．３．３　24709-1で説明している内容

24709-1は、BioAPI（ISO/IEC　19784-1を参照）に適合していることを宣言するバイオメトリック製品の適合性評価を試験するために要求された　概念・フレームワーク・試験方法・基準を説明している。

一揃いのBioAPI適合性評価試験は、3つの立証モデルに対して適合性評価試験を実行するための物である。そして7章から10章では主張言語で記述されている。

このBioAPI適合性評価試験ソフトウエアの形式や構造に関しての制限は特にない。

BioAPI適合性評価試験ソフトウエアは、各試験される実装において、テストログ（１１節参照）を生成しテストレポート（12節参照）を生成する。 もしBioAPI適合性評価試験ソフトウエアが、基本となる標準の実装の試験を実行できなければ、テストレポートにおいて記録される。

３．３．４　適合性評価試験方法

適合性評価試験は、BioAPI実装の機能性の「ブラックボックス」試験である。候補となる実装の内部構造もソース・コードも検査されない。

仕様に適合した入力と適合しない入力の様々な組合せで試験され、生じる出力を1セットの対応する「予期された結果」と比較する。試験結果は、通過/失敗　という基準を使って評価される。

標準BioAPIコンポーネントの各々を別々に対応する。3つの適合試験モデルが、方法論(個々の標準BioAPIコンポーネントの試験用の一つ)の中で定義される。以下に3つのコンポーネントを示す。

a) BioAPIアプリケーションの実装に与えられた標準BioAPIコンポーネント

b) BioAPIフレームワークの実装に与えられた標準BioAPIコンポーネント

c) BioAPI　BSPの　実装に与えられた標準BioAPIコンポーネント

適合性評価試験モデル

適合性評価試験モデルを3つに分け説明している。

a)　BioAPIアプリケーションに対する適合性評価試験モデル

試験コンポーネント(「アプリケーションテストフレームワーク」と呼ぶ)は、試験対象のアプリケーションと基準と

なるフレームワークの間に挿入される。

b)　BioAPIフレームワークのための適合性評価試験モデル

試験コンポーネント(「フレームワーク試験アプリケーション」と呼ぶ)は、基準となるアプリケーションを置き換え

別の試験コンポーネント(「フレームワーク試験BSP」と呼ぶ)は、基準となるBSPを置き換える。

これら2つの試験するコンポーネントの間に、対象となる試験下でフレームワークが存在する。

c)　BioAPI BSPのための適合性評価試験モデル

試験コンポーネント(「BSP試験アプリケーション」と呼ぶ)は、基準となるアプリケーションおよび基準

となる

フレームワークを置き換える。この試験するコンポーネントは、BioAPIアプリケーション、およびBioAPIフレームワークとして両方を演じるものとし、フレームワーク・コールバック・インターフェースを実装する。

３．３．５　主張言語

BioAPI 適合テストの中で使用される主張言語について多くのページを割いて記述する。

主張言語は方法論をテストする BioAPI 適合の不可欠な部分である。W3C XML 1.0 に基づくシンタックスである。

主張言語の要素

本書で説明している要素の一覧を以下の表に列挙する。

8.1	Element <package>	8.11	Element <output> (child of <activity>)
8.2	Element <assertion> (child of <package>)	8.12	Element <set>
8.3	Element <input> (child of <assertion>)	8.13	Element <add>
8.4	Element <invoke> (child of <assertion>)	8.14	Element <subtract>
8.5	Element <input> (child of <invoke>)	8.15	Element <invoke> (child of <activity>)
8.6	Element <output> (child of <invoke>)	8.16	Element <only if>
8.7	Element <return> (child of <invoke>)	8.17	Element <wait until>
8.8	Element <bind> (child of <assertion>)	8.18	Element <assert condition>
8.9	Element <activity> (child of <package>)	8.19	Element <and>
8.10	Element <input> (child of <activity>)	8.20	Element <or>

8.21	Element <xor>
8.22	Element <not>
8.23	Element <equal to>
8.24	Element <not equal to>
8.25	Element <greater than>
8.26	Element <greater than or equal to>
8.27	Element <less than>
8.28	Element <less than or equal to>
8.29	Element <same as>
8.30	Element <not same as>
8.31	Element <existing>
8.32	Element <not existing>

３．３．６　標準インターフェース

標準BioAPIインターフェース機能の使用を記述する。

以下の表に列挙する各関数について、次の項目について説明される

1) 関数呼び出しスキーム

2) パラメーター上の制約;

3) 関数呼び出し入力

4) 関数呼び出し出力

5) 束縛された行動実施の入力

6）束縛された行動実施の出力

7）初期出力

		9.11	BioAPI_QueryUnits
		9.12	BioAPI_EnumBFPs
9.3	BioAPI_Init..	9.13	BioAPI_QueryBFPs
9.4	BioAPI_Terminate	9.14	BioAPI_ControlUnit
9.5	BioAPI_GetFrameworkInfo	9.15	BioAPI_FreeBIRHandle
9.6	BioAPI_EnumBSPs	9.16	BioAPI_GetBIRFromHandle
9.7	BioAPI_BSPLoad	9.17	BioAPI_GetHeaderFromHandle
9.8	BioAPI_BSPUnload	9.18	BioAPI_EnableEvents
9.9	BioAPI_BSPAttach	9.19	BioAPI_SetGUICallbacks
9.10	BioAPI_BSPDetach	9.20	BioAPI_Capture

9.21	BioAPI_CreateTemplate	9.31	BioAPI_DbOpen
9.22	BioAPI_Process	9.32	BioAPI_DbClose
9.23	BioAPI_ProcessWithAuxBIR	9.33	BioAPI_DbCreate
9.24	BioAPI_VerifyMatch	9.34	BioAPI_DbDelete
9.25	BioAPI_IdentifyMatch	9.35	BioAPI_DbSetMarker
9.26	BioAPI_Enroll	9.36	BioAPI_DbFreeMarker
9.27	BioAPI_Verify	9.37	BioAPI_DbStoreBIR
9.28	BioAPI_Identify	9.38	BioAPI_DbGetBIR
9.29	BioAPI_Import	9.39	BioAPI_DbGetNextBIR
9.30	BioAPI_PresetIdentifyPopulation	9.40	BioAPI_DbDeleteBIR

9.41	BioAPI_SetPowerMode	9.51	BioSPI_BSPAttach
9.42	BioAPI_SetIndicatorStatus	9.52	BioSPI_BSPDetach
9.43	BioAPI_GetIndicatorStatus	9.53	BioSPI_QueryUnits
9.44	BioAPI_CalibrateSensor	9.54	BioSPI_QueryBFPs
9.45	BioAPI_Cancel	9.55	BioSPI_ControlUnit
9.46	BioAPI_Free	9.56	BioSPI_FreeBIRHandle
9.47	BioAPI_Util_InstallBSP	9.57	BioSPI_GetBIRFromHandle
9.48	BioAPI_Util_InstallBFP	9.58	BioSPI_GetHeaderFromHandle
9.49	BioSPI_BSPLoad	9.59	BioSPI_EnableEvents
9.50	BioSPI_BSPUnload	9.60	BioSPI_SetGUICallbacks

9.61	BioSPI_Capture	9.71	BioSPI_PresetIdentifyPopulation
9.62	BioSPI_CreateTemplate	9.72	BioSPI_DbOpen
9.63	BioSPI_Process	9.73	BioSPI_DbClose
9.64	BioSPI_ProcessWithAuxBIR	9.74	BioSPI_DbCreate
9.65	BioSPI_VerifyMatch	9.75	BioSPI_DbDelete
9.66	BioSPI_IdentifyMatch	9.76	BioSPI_DbSetMarker
9.67	BioSPI_Enroll	9.77	BioSPI_DbFreeMarker
9.68	BioSPI_Verify	9.78	BioSPI_DbStoreBIR
9.69	BioSPI_Identify	9.79	BioSPI_DbGetBIR
9.70	BioSPI_Import	9.80	BioSPI_DbGetNextBIR

9.81	BioSPI_DbDeleteBIR	9.91	BioSPI_EventHandler
9.82	BioSPI_SetPowerMode	9.92	BioSPI_GUI_STATE_CALLBACK
9.83	BioSPI_SetIndicatorStatus	9.93	BioSPI_GUI_STREAMING_CALLBACK
9.84	BioSPI_GetIndicatorStatus	9.94	BioSPI_BFP_ENUMERATION_HANDLER
9.85	BioSPI_CalibrateSensor	9.95	BioSPI_MEMORY_FREE_HANDLER
9.86	BioSPI_Cancel		
9.87	BioSPI_Free		
9.88	BioAPI_EventHandler		
9.89	BioAPI_GUI_STATE_CALLBACK		
9.90	BioAPI_GUI_STREAMING_CALLBACK		

３．３．７　適合性評価試験の具体的な進め方

一揃いのバイオAPI適合性評価試験(CTS)は、実行可能な試験ケース、関連するテストデータおよびテスト手順の組み合わせから構成される。テストケースは、以下の3項目から構成される。

1）テスト目的の記述(つまり何がテストされているかー 特別のテストによって取り組まれることになっている条件、必要条件あるいは能力)
2）テストケースに対する、パス/失敗 の基準。
3）テストケースが由来するBioAPI明細標準中の節への言及

適合化試験の表決は、下記のうちの一つと結論付けられる。
a) Pass(完了）は、テストケースのテスト目的が集中する一致必要条件との適合性に関する証拠が観察されたということを意味する。
b) Fail(不成功）は、観測されたテスト結果が少なくともテストケースのテスト目的が集中している一致要件の1つに関して非の一致を示すか、または少なくとも1回の無効のテストイベントを含むのを意味します。
d) Inconclusive(未定）または Undecided は、観測されたテスト結果が Pass または fail 評価も与えることができないようなものであることを意味します。

３．４　ISO／IEC24709－2:2007 Information technology－Conformance Testing for BioAPI－Part2:Test Assertion for Biometric Service Providersについて

３．４．１　はじめに

本国際規格では、ＩＳＯ／ＩＥＣ１９７８４－１：２００６への準拠性を試験するため、ＢｉｏＡＰＩの基本構成要素であるＢＳＰ（バイオメトリック・サービス・プロバイダ）の製品に関する適合試験方法を定義している。以下にＩＳＯ／ＩＥＣ２４７０９－２の内容とアサーション言語（ＸＭＬ表記法）による適合性判定の試験仕様を翻訳、引用しながらその骨子をまとめる。（適合性判定の試験仕様の内容は付録を参照のこと）

３．４．２　適用範囲

本規格ではＩＳＯ／ＩＥＣ２４７０９－１：２００７の方法論に基づきＢＳＰ実装体の適合評価方法を定義している。１９７８４－１で定義されているＢＳＰのサブクラス全てに関しての試験詳細が定義されている。（詳細は、３．３．６（１）節を参照）

３．４．３　適合性について

本国際規格は、ＩＳＯ／ＩＥＣ２４７０９：２００７のパート１にて定義された適合性試験用に定義されたＢｉｏＡＰＩ２．０適合性試験の方法論および適合性試験の詳細定義言語（アサーション言語）によりＢＳＰ関数の適合試験の判定方法を定義している。本規格に適合するＢｉｏＡＰＩ適合試験スイート（適合試験プログラム）の実装実体は、ＩＳＯ／ＩＥＣ２４７０９－１：２００７で定義された方法論と本規格書８節（本書の付録を参照）の定義に適合しなければならない。

３．４．４　引用文献

ISO／IEC19784－1:2006 Information technology － Biometric application programming interface － Part 1:BioAPI specification

ISO/IEC24709-1:2007 Information technology - Conformance testing for the biometric application programming interface(BioAPI) － Part 1:Methods and Procedures

３．４．５　一般原則

３．３．６章にリスト化されている適合評価試験定義文（試験アサーション）は、ＩＳＯ／ＩＥＣ２４７０９－１：２００７で定義された適合性評価試験の方法論を基に３．３．６節で定義されており、この方法論の内容に専ら従い使用することができる。ＩＳＯ／ＩＥＣ２４７０９－１定義された適合性試験方法論の重要な概念として、３つの適合性試験モデルが存在することである。（ＩＳＯ／ＩＥＣ２４７０９－１：２００７、６章参照）

１）ＢｉｏＡＰＩアプリケーションに対する適合性試験モデル

２）ＢｉｏＡＰＩフレームワークに対する適合性試験モデル

３）ＢｉｏＡＰＩＢＳＰに対する適合性試験モデル

これらの試験モデルは、各々ＢｉｏＡＰＩ構造において３つの標準構成要素の中の１つに関連している。本文の３．３．７章には、ＡＰＩ準拠ＢＳＰに対する適合性評価試験で実施される多くの試験アサーションが定義されている。本規格文書では、ＢｉｏＡＰＩ　Ａｐｐｌｉｃａｔｉｏｎ　及びフレームワークの適合性試験モデルについては取り扱わない。

３．４．６　ＢｉｏＡＰＩＢＳＰの適合性評価試験（テストアサーション）

（１）　一般事項

ＩＳＯ／ＩＥＣ２４７０９－１：２００７規格書の６．１節で定義されているように、Ｂｉｏ

ＡＰＩ準拠ＢＳＰの適合性試験に対して、「ＢｉｏＡＰＩ適合一覧表（ＢｉｏＡＰＩ　Ｃｏｎｆｏｒｍａｎｃｅ　Ｓｔａｔｅｍｅｎｔ）」が作成されなければならない。

表３－８ ―BioAPI適合一覧表

Vendor contact information:	
Name	
Address:	
Street	
City	
State or province	
Zip code or postal code	
Country	
Telephone	
Biometric product:	
Name	
Serial number	
Description	
BioAPI conformance class (one of the following):	
BioAPI application	
BioAPI framework	
BSP (this alternative shall be selected)	**X**
Conformance subclass of the BSP (one of the following):	
verification BSP	
identification BSP	
capture BSP	
verification engine	
identification engine	
Optional functions supported by the BSP (zero or more of the following):	
BioSPI_Capture	
BioSPI_CreateTemplate	
BioSPI_Process	
BioSPI_VerifyMatch	
BioSPI_Enroll	
BioSPI_Verify	
BioSPI_Identify	
Optional features supported by the BSP (zero or more of the following):	
BSP-controlled database	
template adaptation	
generation of SOURCE PRESENT events	
setting of the Quality field in an intermediate BIR	
setting of the Quality field in a processed BIR	
Additional information:	
UUID of the BSP	
BioAPI specification version	
Product version	

Supported BDB formats (one or more format owner / format type pairs)	
Supported biometric types (one or more)	
Maximum supported size for the payload	

次の表３－９は、試験対象の実装体（ＢｉｏＡＰＩ準拠ＢＳＰ）がＩＳＯ／ＩＥＣ１９７８４－１：２００６で定義された要件を満足しているかどうかを判定するために使用される試験アサーションをリスト化したものである。

注記）実装体が、利用可能な試験アサーションを順次、成功裏に処理されることがＩＳＯ／ＩＥＣ１９７８４－１：２００６の利用可能な要件を満たしているということの明確な主たる証拠となる。しかし、これらのアサーションが適合試験を網羅したものではないため（網羅できないため）、適合性の確証ではない。（ＩＳＯ／ＩＥＣ２４７０９－１：２００７、６節参照）

これらのアサーションはＢｉｏＡPI準拠ＢＳＰに対する適合性評価試験モデルを使用する。

以下の付属節には、ＢＳＰの付属クラスの各々の適合評価のために用意すべきものが含まれる。そして与えらた付属クラスのＢＳＰに対する適合評価試験を実行するために定義された試験アサーションが規定されている。

表３－９— Test assertions for BioAPI BSP

Number	Assertion name	References	Purpose
1a	*BioSPI_BSPLoad_InvalidUUID*	**9.3.1.1, 11.2.3**	Verify if calling the function BioSPI_BSPLoad with an invalid input parameter UUID returns BioAPIERR_H_FRAMEWORK_INVALID_UUID.
1b	*BioSPI_BSPLoad_ValidParam*	**9.3.1.1, A.4**	Verify if calling BioSPI_BSPLoad with valid input parameters returns BioAPI_OK.
2a	*BioSPI_BSPUnload_ValidParam*	**9.3.1.2, A.4**	Verify if calling BioSPI_BSPUnload with valid input parameters returns BioAPI_OK.
2b	*BioSPI_BSPUnload_InvalidUUID*	**9.3.1.2, 11.2.3**	Verify if calling BioSPI_BSPUnload with an invalid input parameter UUID returns BioAPIERR_INVALID_UUID.
2c	*BioSPI_BSPUnload_UnmatchedLoad*	**9.3.1.2, 8.1.6.1**	Verify if calling BioSPI_BSPUnload without a matching call to BioSPI_BSPLoad returns BioAPIERR_BSP_NOT_LOADED.
2d	*BioSPI_BSPUnload_Confirm*	**9.3.1.2**	Verify if calling BioSPI_BSPUnload truly unloads the BSP.
3a	*BioSPI_BSPAttach_ValidParam*	**9.3.1.3, A.4**	Verify if a call to the function BioSPI_BSPAttach with valid input parameters returns BioAPI_OK.
3b	*BioSPI_BSPAttach_InvalidUUID*	**9.3.1.3, 11.2.3**	Verify if a call to the function BioSPI_BSPAttach returns BioAPIERR_INVALID_UUID when

			the input UUID is invalid.
3c	*BioSPI_BSPAttach_InvalidVersion*	**9.3.1.3, 11.2.3**	Verify if a call to the function BioSPI_BSPAttach returns BioAPIERR_INCOMPATIBLE_VERSION when the caller specifies an incompatible version.
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	**9.3.1.3**	Verify if the function BioSPI_BSPAttach returns an error when called with an invalid BSP handle.
4a	*BioSPI_BSPDetach_ValidParam*	**9.3.1.4, A.4**	Verify if a call to the function BioSPI_BSPDetach with a valid module handle returns BioAPI_OK.
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	**9.3.1.4, 8.1.8.1**	Verify if a call to BioSPI_BSPDetach with an invalid module handle returns an error.
4c	*BioSPI_BSPDetach_Confirm*	**9.3.1.4, 8.1.8.4**	Verify if a call to the function BioSPI_BSPDetach truly terminates the attach session.
5a	*BioSPI_FreeBIRHandle_ValidParam*	**9.3.2.1, 8.2.1, A.4**	Verify if calling the function BioSPI_FreeBIRHandle with a valid BIR handle frees the BIR handle.
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	**9.3.2.1, 8.2.1**	Verify if calling BioSPI_FreeBIRHandle with an invalid BSP handle returns an error.
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	**9.3.2.1, 8.2.1**	Verify if calling BioSPI_FreeBIRHandle with an invalid BIR handle returns an error.
6a	*BioSPI_GetBIRFromHandle_ValidParam*	**9.3.2.2, 8.2.2, A.4**	Verify if calling BioSPI_GetBIRFromHandle with valid parameters returns BioAPI_OK.
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	**9.3.2.2, 8.2.2**	Verify if calling BioSPI_GetBIRFromHandle with an invalid BSP handle returns an error.
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	**9.3.2.2, 8.2.2**	Verify if calling the function BioSPI_GetBIRFromHandle with an invalid BIR handle returns an error.
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	**9.3.2.3, 8.2.3, A.4**	Verify if a call to the function BioSPI_GetHeaderFromHandle with valid input parameters returns BioAPI_OK.
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	**9.3.2.3, 8.2.3**	Verify if invoking the function BioSPI_GetHeaderFromHandle with an invalid module handle returns an error.
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	**9.3.2.3, 8.2.3**	Verify if a call to the function BioSPI_GetHeaderFromHandle with an invalid BIR handle returns an error.
7d	*BioSPI_GetHeaderFromHandle_BIRHandleNotFre*	**9.3.2.3,**	Verify that after a call to the

	ed	**8.2.3**	function BioSPI_GetHeaderFromHandle, the BIR handle has not been freed.
8a	*BioSPI_EnableEvents_ValidParam*	**9.3.3.1, 8.3.1**	Verify BioSPI_EnableEvents with valid input parameters.
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	**9.3.3.1, 8.3.1**	Verify BioSPI_EnableEvents with an invalid module handle.
9a	*BioSPI_Capture_AuditData*	**9.3.4.1, 8.4.1.1, 7.47, A.4.6.2.1**	Verify the function BioSPI_Capture with AuditData having a non-NULL value.
9b	*BioSPI_Capture_ReturnQuality*	**9.3.4.1, 8.4.1, 7.49, 7.47, A.4.6.2.2**	Verify if the header of the captured BIR contains a valid quality value (in the range 0-100).
9c	*BioSPI_Capture_IntermediateProcessedBIR*	**9.3.4.1, 8.4.1.1, 8.4.1.2**	Verify if the BIR returned by the function BioSPI_Capture has a processed level of either INTERMEDIATE or PROCESSED.
9d	*BioSPI_Capture_InvalidBSPHandle*	**9.3.4.1, 8.4.1.2**	Verify if calling BioSPI_Capture with an invalid BSP handle returns an error BioAPIERR_INVALID_BSP_HANDLE.
10a	*BioSPI_CreateTemplate_PayloadSupported*	**9.3.4.2, 8.4.5.1, 7.47, A.4.6.2.6**	Verify BioSPI_CreateTemplate with valid parameters and payload.
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	**9.3.4.2, 7.49.3, 7.47, A.4.6.2.2**	Verify the function BioSPI_CreateTemplate with valid parameters and the returned quality value.
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	**9.3.4.2, 8.4.2.1**	Verify BioSPI_CreateTemplate with valid parameters. The new template BIR is expected to have the processed level PROCESSED.
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	**9.3.4.2, 8.4.2.1**	Verify if the purpose of the created template is the same as the purpose of the captured BIR.
10e	*BioSPI_CreateTemplate_InputBIRDataType*	**9.3.4.2, 8.4.2.1**	Verify BioSPI_CreateTemplate with an input BIR that has an invalid processed level.
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	**9.3.4.2, 8.4.2.1**	Verify function BioSPI_CreateTemplate with an invalid input BIR purpose.
11a	*BioSPI_Process_ValidParam*	**9.3.4.3, 8.4.3.1**	Verify BioSPI_Process with valid parameters and the returned processed level.
11b	*BioSPI_Process_BIRHeaderQuality*	**9.3.4.3, 8.4.3.1, 7.49.3, 7.47, A.4.6.2.2**	Verify BioSPI_Process with valid parameters and if the returned quality is valid.

11c	*BioSPI_Process_OutputBIRPurpose*	**9.3.4.3, 8.4.3.1, 7.12.3**	Verify BioSPI_Process with valid parameters and if the purpose of the processed BIR is the same as the purpose of the captured BIR.
11d	*BioSPI_Process_BuildsProcessedBIR*	**9.3.4.3, 8.4.3.1**	Verify BioSPI_Process with valid parameters and the returned processed level.
11e	*BioSPI_Process_InputBIRDataType*	**9.3.4.3, 8.4.3.1**	Verify BioSPI_Process with an input BIR having a processed level of PROCESSED and if the BioSPI_Process call fails.
12a	*BioSPI_VerifyMatch_ValidParam*	**9.3.4.5, 8.4.5.1**	Verify if calling BioSPI_VerifyMatch with valid input parameters returns BioAPI_OK.
12b	*BioSPI_VerifyMatch_Payload*	**9.3.4.5, 8.4.5.1, 7.47**	Verify the support of payload in the function BioSPI_VerifyMatch. The function is expected to return BioAPI_OK.
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	**9.3.4.5, 8.4.5.1, 11.2.3**	Verify BioSPI_VerifyMatch with a BIR whose purpose is invalid for the function. The function is expected to return BioAPIERR_BSP_INCONSISTENT_PURPOSE.
13a	*BioSPI_Enroll_ValidParam*	**9.3.4.7, 8.4.7.1, A.4**	Verify if calling BioSPI_Enroll with valid input parameters returns BioAPI_OK.
13b	*BioSPI_Enroll_Payload*	**9.3.4.7, 8.4.7.1, A.4.6.2.6, 7.47**	Verify if calling BioSPI_Enroll with a payload returns BioAPI_OK.
13c	*BioSPI_Enroll_AuditData*	**9.3.4.7, 8.4.7.1, 7.47, A.4.6.2.1**	Verify BioSPI_Enroll with a non-NULL AuditData parameter. The function is expected to return audit data if the BSP supports audit data.
13d	*BioSPI_Enroll_BIRHeaderQuality*	**9.3.4.7, 8.4.7.1, 7.49.3, 7.47, A.4.6.2.2**	Verify if calling the function BioSPI_Enroll returns a valid quality value in the new template BIR's header.
14a	*BioSPI_Verify_ValidParam*	**9.3.4.8, 8.4.8.1, A.4**	Verify if calling the function BioSPI_Verify with valid input parameters returns BioAPI_OK.
14b	*BioSPI_Verify_Payload*	**9.3.4.8, 8.4.8.1, 7.47, A.4.6.2.6**	Verify if calling the function BioSPI_Verify with a payload returns BioAPI_OK.
14c	*BioSPI_Verify_AuditData*	**9.3.4.8, 8.4.8.1, 7.47, A.4.6.2.1**	Verify if calling BioSPI_Verify with audit data returns BioAPI_OK.
15a	*BioSPI_DbOpen_ValidParam*	**9.3.5.1, 8.5.1.1**	Verify BioSPI_DbOpen with valid input parameters

15b	*BioSPI_DbOpen_InvalidBSPHandle*	**9.3.5.1, 8.5.1.1**	Verify BioSPI_DbOpen with an invalid BSP handle
16a	*BioSPI_DbClose_ValidParam*	**9.3.5.2, 8.5.2.1**	Verify BioSPI_DbClose with valid input parameters
16b	*BioSPI_DbClose_InvalidBSPHandle*	**9.3.5.2, 8.5.2.1**	Verify BioSPI_DbClose with an invalid BSP handle
17a	*BioSPI_DbCreate_DbProtected*	**9.3.5.3, 8.5.3.1**	Invoke BioSPI_DbCreate twicw, and verify if the second invocation results in returning error code BioAPIERR_DATABASE_ALREADY_EXISTS.
17b	*BioSPI_DbCreate_ValidParam*	**9.3.5.3, 8.5.3.1**	Verify BioSPI_DbCreate with valid input parameters
17c	*BioSPI_DbCreate_InvalidBSPHandle*	**9.3.5.3, 8.5.3.1**	Verify BioSPI_DbCreate with invalid BSP handle
18a	*BioSPI_DbDelete_InvalidBSPHandle*	**9.3.5.4, 8.5.4.1**	Verify BioSPI_DbDelete with an invalid BSP handle
18b	*BioSPI_DbDelete_OpenDbProtected*	**9.3.5.4, 8.5.4.1**	Verify BioSPI_DbDelete when the database is open
18c	*BioSPI_DbDelete_ValidParam*	**9.3.5.4, 8.5.4.1**	Verify BioSPI_DbDelete with valid input parameters
19a	*BioSPI_DbSetMarker_ValidParam*	**9.3.5.5, 8.5.5.1**	Verify BioSPI_DbSetMarker with valid input parameters
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	**9.3.5.5, 8.5.5.1**	Verify BioSPI_DbSetMarker with invalid BSP handle
19c	*BioSPI_DbSetMarker_RecordNotFound*	**9.3.5.5, 8.5.5.1**	Verify BioSPI_DbSetMarker with a keyvalue that does not exist in the database
20a	*BioSPI_DbFreeMarker_ValidParam*	**9.3.5.6, 8.5.6.1**	Verify BioSPI_DbFreeMarker with valid input parameters
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	**9.3.5.6, 8.5.6.1**	Verify BioSPI_DbFreeMarker with an invalid BSP handle
20c	*BioSPI_DbFreeMarker_InvalidMarker*	**9.3.5.6, 8.5.6.1**	Verify BioSPI_DbFreeMarker with invalid marker handle
21a	*BioSPI_DbStoreBIR_ValidParam*	**9.3.5.7, 8.5.7.1**	Verify BioSPI_DbStoreBIR with valid input parameters
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	**9.3.5.7, 8.5.7.1**	Verify BioSPI_DbStoreBIR with invalid BSP handle
22a	*BioSPI_DbGetBIR_ValidParam*	**9.3.5.8, 8.5.8.1**	Verify BioSPI_DbGetBIR with valid input parameters
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	**9.3.5.8, 8.5.8.1**	Verify BioSPI_DbGetBIR with invalid BSP handle
22c	*BioSPI_DbGetBIR_RecordNotFound*	**9.3.5.8, 8.5.8.1**	Verify BioSPI_DbGetBIR with a key value that does not exist in the database
23a	*BioSPI_DbGetNextBIR_ValidParam*	**9.3.5.9, 8.5.9.1**	Verify BioSPI_DbGetNextBIR with valid input parameters

23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	**9.3.5.9, 8.5.9.1**	Verify BioSPI_DbGetNextBIR with invalid BSP handle
24a	*BioSPI_DbDeleteBIR_ValidParam*	**9.3.5.10, 8.5.10.1**	Verify BioSPI_DbDeleteBIR with valid parameters
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	**9.3.5.10, 8.5.10.1**	Verify BioSPI_DbDeleteBIR with an invalid BSP handle

（2） 付属クラス「照合ＢＳＰ」に属するＢＳＰの試験

適合性の分類「照合ＢＳＰ」に含まれるＢＳＰが、ＩＳＯ／ＩＥＣ１９７８４－１：２００６付録Ａ．４及びＡ４．１で定義された要件を満足するかどうかを判定するために使用する試験アサーションが以下に定義されている。

全ての照合ＢＳＰは、次に示すアサーションを順に全て実行し試験されなければならない。

表３－９ — Assertions for Testing Verification BSP

Number	Assertion name	Package
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e
4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b

13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b
14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿Ｃａｐｔｕｒｅ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１０— Assertions for Testing Verification BSP - BioSPI_Capture

Number	Assertion name	Package
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿ＣｒｅａｔｅＴｅｍｐｌａｔｅ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１１ — Assertions for Testing Verification BSP - BioSPI_CreateTemplate

Number	Assertion name	Package
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿Ｐｒｏｃｅｓｓ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１２— Assertions for Testing Verification BSP - BioSPI_Process

Number	Assertion name	Package
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b
11d	*BioSPI_Process_BuildsProcessedBIR*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿ＶｅｒｉｆｙＭａｔｃｈ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１３— Assertions for Testing Verification BSP - BioSPI_VerifyMatch

Number	Assertion name	Package
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

もし、試験対象の実装体が、ＢＳＰ制御のデータベース（１９７８４－１：２００６、付録Ａ．４．６．１（２）参照）のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１４— Assertions for Testing Verification BSP - Database Functions

Number	Assertion name	Package
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbClose_ValidParam*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b
17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
24a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

（３）　付属クラス「識別ＢＳＰ」に属するＢＳＰの試験

適合性の分類「識別ＢＳＰ」に含まれるＢＳＰが、ＩＳＯ／ＩＥＣ１９７８４－１：２００６付録Ａ．２及びＡ４．２で定義された要件を満足するかどうかを判定するために使用する試験アサーションが以下に定義されている。
全ての識別ＢＳＰは、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１５ — Assertions for Testing Identification BSP

Number	Assertion name	Package
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e
4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
7d	*BioSPI_GetHeaderFromHandle_BIRHandleNotFreed*	01cc0988-0ccf-1085-a367-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b
13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b
14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿Ｃａｐｔｕｒｅ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１６ — Assertions for Testing Identification BSP - BioSPI_Capture

Number	Assertion name	Package
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿ＣｒｅａｔｅＴｅｍｐｌａｔｅ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１７— Assertions for Testing Identification BSP - BioSPI_CreateTemplate

Number	Assertion name	Package
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿Ｐｒｏｃｅｓｓ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１８— Assertions for Testing Identification BSP - BioSPI_Process

Number	Assertion name	Package
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b
11d	*BioSPI_Process_BuildsProcessedBIR*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿ＶｅｒｉｆｙＭａｔｃｈ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－１９— Assertions for Testing Identification BSP - BioSPI_VerifyMatch

Number	Assertion name	Package
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b

12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

もし、試験対象の実装体が、ＢＳＰ制御のデータベース（１９７８４－１：２００６、付録Ａ．４．６．１（２）参照）のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２０ ― Assertions for Testing Identification BSP - Database Functions

Number	Assertion name	Package
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbClose_ValidParam*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b
17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
24a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

（４） 付属クラス「キャプチャＢＳＰ」に属するＢＳＰの試験

適合性の分類「キャプチャＢＳＰ」に含まれるＢＳＰが、ＩＳＯ／ＩＥＣ１９７８４－１：２０

０６付録A．４及びA４．３で定義された要件を満足するかどうかを判定するために使用する試験アサーションが以下に定義されている。
この付属クラスの全ＢＳＰは、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２１ — Assertions for Testing Capture BSP

Number	Assertion name	Package
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e
4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
7d	*BioSPI_GetHeaderFromHandle_BIRHandleNotFreed*	01cc0988-0ccf-1085-a367-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

もし、試験対象の実装体が、ＢｉｏＳＰＩ__ＣｒｅａｔｅＴｅｍｐｌａｔｅ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２２ **— Assertions for Testing Capture BSP - BioSPI_CreateTemplate**

Number	Assertion name	Package
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿Ｐｒｏｃｅｓｓ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２３ **— Assertions for Testing Capture BSP - BioSPI_Process**

Number	Assertion name	Package
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b
11d	*BioSPI_Process_BuildsProcessedBIR*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿ＶｅｒｉｆｙＭａｔｃｈ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２４ **— Assertions for Testing Capture BSP - BioSPI_VerifyMatch**

Number	Assertion name	Package
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿Ｅｎｒｏｌｌ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２５ **— Assertions for Testing Capture BSP - BioSPI_Enroll**

Number	Assertion name	Package
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b
13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿Ｖｅｒｉｆｙ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２６ **— Assertions for Testing Capture BSP - BioSPI_Verify**

Number	Assertion name	Package
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b

14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

もし、試験対象の実装体が、ＢＳＰ制御のデータベース（１９７８４－１：２００６、付録Ａ．４．６．１．２参照）のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２７― Assertions for Testing Capture BSP - Database Functions

Number	Assertion name	Package
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbClose_ValidParam*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b
17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
24a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

（５） 付属クラス「照合エンジンＢＳＰ」に属するＢＳＰの試験

適合性の分類「照合エンジンＢＳＰ」に含まれるＢＳＰが、ＩＳＯ／ＩＥＣ１９７８４－１：２００６付録Ａ．４及びＡ４．４で定義された要件を満足するかどうかを判定するために使用する

試験アサーションが以下に定義されている。
この付属クラスの全ＢＳＰは、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２８ — Assertions for Testing Verification Engine BSP

Number	Assertion name	Package
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e
4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
7d	*BioSPI_GetHeaderFromHandle_BIRHandleNotFreed*	01cc0988-0ccf-1085-a367-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b

11d	*BioSPI_Process_BuildsProcessedBIR*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

もし、試験対象の実装体が、ＢｉｏＳＰＩ__Ｃａｐｔｕｒｅ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－２９ — Assertions for Testing Verification Engine BSP - BioSPI_Capture

Number	Assertion name	Package
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

もし、試験対象の実装体が、ＢｉｏＳＰＩ__Ｅｎｒｏｌｌ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－３０ — Assertions for Testing Verification Engine BSP - BioSPI_Enroll

Number	Assertion name	Package
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b
13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ__Ｖｅｒｉｆｙ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－３１— Assertions for Testing Verification Engine BSP - BioSPI_Verify

Number	Assertion name	Package
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b
14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

もし、試験対象の実装体が、ＢＳＰ制御のデータベース（１９７８４－１：２００６、付録Ａ．４．６．１（２）参照）のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－３２ — Assertions for Testing Verification Engine BSP - Database Functions

Number	Assertion name	Package
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbClose_ValidParam*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b

17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b
21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
24a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

（6） 付属クラス「識別エンジンＢＳＰ」に属するＢＳＰの試験

適合性の分類「識別エンジンＢＳＰ」に含まれるＢＳＰが、ＩＳＯ／ＩＥＣ１９７８４－１：２００６付録Ａ．４及びＡ４．５で定義された要件を満足するかどうかを判定するために使用する試験アサーションが以下に定義されている。

この付属クラスの全ＢＳＰは、次に示すアサーションを順に全て実行し試験されなければならない。

表３－３３ — Assertions for Testing Identification Engine BSP

Number	Assertion name	Package
1a	*BioSPI_BSPLoad_InvalidUUID*	020e90c8-0c19-1085-ab54-0002a5d5fd2e
1b	*BioSPI_BSPLoad_ValidParam*	01f6c6f0-0c19-1085-97fe-0002a5d5fd2e
2a	*BioSPI_BSPUnload_ValidParam*	01661010-0c22-1085-8688-0002a5d5fd2e
2b	*BioSPI_BSPUnload_InvalidUUID*	01c2e5b0-0c3b-1085-b31d-0002a5d5fd2e
2c	*BioSPI_BSPUnload_UnmatchedLoad*	02f6c618-0c23-1085-ba89-0002a5d5fd2e
2d	*BioSPI_BSPUnload_Confirm*	03daf040-0c3b-1085-a9fd-0002a5d5fd2e
3a	*BioSPI_BSPAttach_ValidParam*	00ae6488-0c3d-1085-9912-0002a5d5fd2e
3b	*BioSPI_BSPAttach_InvalidUUID*	049cc170-0c5f-1085-981f-0002a5d5fd2e
3c	*BioSPI_BSPAttach_InvalidVersion*	0052ac10-0c60-1085-9883-0002a5d5fd2e
3d	*BioSPI_BSPAttach_InvalidBSPHandle*	03826830-0c57-1085-bfb0-0002a5d5fd2e

4a	*BioSPI_BSPDetach_ValidParam*	00e0d2b0-0c7a-1085-b8ac-0002a5d5fd2e
4b	*BioSPI_BSPDetach_InvalidBSPHandle*	0434c458-0c79-1085-9f2c-0002a5d5fd2e
4c	*BioSPI_BSPDetach_Confirm*	002e7e58-0c78-1085-9e1d-0002a5d5fd2e
5a	*BioSPI_FreeBIRHandle_ValidParam*	0280a7d0-0c80-1085-a9a0-0002a5d5fd2e
5b	*BioSPI_FreeBIRHandle_InvalidBSPHandle*	047aed48-0c80-1085-898b-0002a5d5fd2e
5c	*BioSPI_FreeBIRHandle_InvalidBIRHandle*	018e6c18-0c9c-1085-afdf-0002a5d5fd2e
6a	*BioSPI_GetBIRFromHandle_ValidParam*	0460b658-0cb4-1085-a304-0002a5d5fd2e
6b	*BioSPI_GetBIRFromHandle_InvalidBSPHandle*	02445668-0cc5-1085-a3ac-0002a5d5fd2e
6c	*BioSPI_GetBIRFromHandle_InvalidBIRHandle*	0194a9c0-0cc7-1085-8780-0002a5d5fd2e
7a	*BioSPI_GetHeaderFromHandle_ValidParam*	027a7db0-0cc7-1085-9391-0002a5d5fd2e
7b	*BioSPI_GetHeaderFromHandle_InvalidBSPHandle*	057e0d38-0ccd-1085-83b8-0002a5d5fd2e
7c	*BioSPI_GetHeaderFromHandle_InvalidBIRHandle*	02195e68-0cce-1085-a46f-0002a5d5fd2e
7d	*BioSPI_GetHeaderFromHandle_BIRHandleNotFreed*	01cc0988-0ccf-1085-a367-0002a5d5fd2e
8a	*BioSPI_EnableEvents_ValidParam*	0333f628-0ccf-1085-aceb-0002a5d5fd2e
8b	*BioSPI_EnableEvents_InvalidBSPHandle*	04ed0838-0ccf-1085-b64e-0002a5d5fd2e
10a	*BioSPI_CreateTemplate_PayloadSupported*	04a01118-0cf9-1085-96d4-0002a5d5fd2e
10b	*BioSPI_CreateTemplate_BIRHeaderQuality*	00b5c728-0cfb-1085-8969-0002a5d5fd2e
10c	*BioSPI_CreateTemplate_OutputBIRDataType*	0193c730-0cf9-1085-b0a3-0002a5d5fd2e
10d	*BioSPI_CreateTemplate_OutputBIRPurpose*	03dbdaa0-0cf2-1085-99ed-0002a5d5fd2e
10e	*BioSPI_CreateTemplate_InputBIRDataType*	6d543ea0-2ce9-11d9-9669-0800200c9a66
10f	*BioSPI_CreateTemplate_Inconsistent_Purpose*	28ec1620-e995-11d9-b1d1-0002a5d5c51b
11a	*BioSPI_Process_ValidParam*	4ec34700-e9a0-11d9-8fc8-0002a5d5c51b
11b	*BioSPI_Process_BIRHeaderQuality*	211668e0-e9a6-11d9-bcc8-0002a5d5c51b
11c	*BioSPI_Process_OutputBIRPurpose*	e1bb4f20-ed61-11d9-9344-0002a5d5c51b
11d	*BioSPI_Process_BuildsProcessedBIR*	f2ce6540-ed66-11d9-9618-0002a5d5c51b
11e	*BioSPI_Process_InputBIRDataType*	3cf96080-ed6b-11d9-9acf-0002a5d5c51b
12a	*BioSPI_VerifyMatch_ValidParam*	688aad60-ee30-11d9-a62c-0002a5d5c51b
12b	*BioSPI_VerifyMatch_Payload*	692ebe20-ee47-11d9-bd34-0002a5d5c51b
12c	*BioSPI_VerifyMatch_Inconsistent_Purpose*	9108ec70-2e9b-11d9-9669-0800200c9a66

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿Ｃａｐｔｕｒｅ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－３４ — Assertions for Testing Identification Engine BSP - BioSPI_Capture

Number	Assertion name	Package
9a	*BioSPI_Capture_AuditData*	02704c50-0cd8-1085-96cb-0002a5d5fd2e
9b	*BioSPI_Capture_ReturnQuality*	03f601f0-0cd8-1085-bd59-0002a5d5fd2e
9c	*BioSPI_Capture_IntermediateProcessedBIR*	055ddb08-0cd6-1085-a6d3-0002a5d5fd2e
9d	*BioSPI_Capture_InvalidBSPHandle*	0244f2a8-0cf2-1085-8443-0002a5d5fd2e

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿Ｅｎｒｏｌｌ関数のサポートを主張している場合、

全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－３５ **— Assertions for Testing Identification Engine BSP - BioSPI_Enroll**

Number	Assertion name	Package
13a	*BioSPI_Enroll_ValidParam*	0b5ebb60-eefb-11d9-990c-0002a5d5c51b
13b	*BioSPI_Enroll_Payload*	e8969d40-ef05-11d9-9098-0002a5d5c51b
13c	*BioSPI_Enroll_AuditData*	b40a5260-ef14-11d9-a4fe-0002a5d5c51b
13d	*BioSPI_Enroll_BIRHeaderQuality*	6f727320-ef1a-11d9-9143-0002a5d5c51b

もし、試験対象の実装体が、ＢｉｏＳＰＩ＿＿Ｖｅｒｉｆｙ関数のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－３６ **— Assertions for Testing Identification Engine BSP - BioSPI_Verify**

Number	Assertion name	Package
14a	*BioSPI_Verify_ValidParam*	b78e5be0-efcb-11d9-b2c7-0002a5d5c51b
14b	*BioSPI_Verify_Payload*	32969ec0-eff8-11d9-9831-0002a5d5c51b
14c	*BioSPI_Verify_AuditData*	89719700-f218-11d9-b028-0002a5d5c51b

もし、試験対象の実装体が、ＢＳＰ制御のデータベース（１９７８４－１：２００６、付録Ａ．４．６．１．２参照）のサポートを主張している場合、全ての試験対象の実装体は、次に示すアサーションを順に全て実行し試験されなければならない。

表３－３７**— Assertions for Testing Identification Engine BSP - Database Functions**

Number	Assertion name	Package
15a	*BioSPI_DbOpen_ValidParam*	e68ff9a0-e506-11d9-a6a1-0002a5d5c51b
15b	*BioSPI_DbOpen_InvalidBSPHandle*	bfd44400-e5de-11d9-bdb9-0002a5d5c51b
16a	*BioSPI_DbClose_ValidParam*	39aa9560-e5f1-11d9-89f3-0002a5d5c51b
16b	*BioSPI_DbClose_InvalidBSPHandle*	6e3f5c00-e5f3-11d9-b663-0002a5d5c51b
17a	*BioSPI_DbCreate_DbProtected*	7b6c2f40-e650-11d9-812f-0002a5d5c51b
17b	*BioSPI_DbCreate_ValidParam*	1421ec38-1db6-49d4-873d-03e2de17598b
17c	*BioSPI_DbCreate_InvalidBSPHandle*	ef4bb862-79f6-4f01-8f5d-af5c3abf23c0
18a	*BioSPI_DbDelete_InvalidBSPHandle*	678e5d12-3d51-41ec-a672-13f34ea24545
18b	*BioSPI_DbDelete_OpenDbProtected*	9e4d0c6d-4d59-479c-9f58-160ecde99aad
18c	*BioSPI_DbDelete_ValidParam*	499d9cc3-4269-4671-9d69-29a31bc1a08f
19a	*BioSPI_DbSetMarker_ValidParam*	94271080-e723-11d9-898c-0002a5d5c51b
19b	*BioSPI_DbSetMarker_InvalidBSPHandle*	69f7ce20-e72e-11d9-b4e0-0002a5d5c51b
19c	*BioSPI_DbSetMarker_RecordNotFound*	b9c90f40-e7ec-11d9-a435-0002a5d5c51b
20a	*BioSPI_DbFreeMarker_ValidParam*	2e1c9520-e7f1-11d9-a011-0002a5d5c51b
20b	*BioSPI_DbFreeMarker_InvalidBSPHandle*	0f735140-e800-11d9-8a8a-0002a5d5c51b
20c	*BioSPI_DbFreeMarker_InvalidMarker*	a9007b60-e802-11d9-8a5d-0002a5d5c51b
21a	*BioSPI_DbStoreBIR_ValidParam*	da953680-e806-11d9-90d3-0002a5d5c51b

21b	*BioSPI_DbStoreBIR_InvalidBSPHandle*	e39027e0-e80b-11d9-85eb-0002a5d5c51b
22a	*BioSPI_DbGetBIR_ValidParam*	67512700-e811-11d9-a5e0-0002a5d5c51b
22b	*BioSPI_DbGetBIR_InvalidBSPHandle*	f62947e0-e8b7-11d9-9dad-0002a5d5c51b
22c	*BioSPI_DbGetBIR_RecordNotFound*	37457440-e8ba-11d9-87da-0002a5d5c51b
23a	*BioSPI_DbGetNextBIR_ValidParam*	e3396400-e8c9-11d9-990f-0002a5d5c51b
23b	*BioSPI_DbGetNextBIR_InvalidBSPHandle*	f0ef5320-e8d3-11d9-bbc1-0002a5d5c51b
24a	*BioSPI_DbDeleteBIR_ValidParam*	ed4afbe0-e8d6-11d9-9ed0-0002a5d5c51b
24b	*BioSPI_DbDeleteBIR_InvalidBSPHandle*	72eca940-e8d9-11d9-aa0d-0002a5d5c51b

３．４．７　適合性評価試験詳細仕様

ＢｉｏＡＰＩ　ＢＳＰ用テストアサーション詳細仕様は
本規格書の８節”ＢｉｏＡＰＩ　ＢＳＰテストアサーション”参照

４　ＢｉｏＡＰＩ２．０フレームワークの適合評価試験仕様

４．１　はじめに

本仕様書は、国際規格ISO19784-1:2006　Information Technology-Biometric application programming interface－ Part1:BioAPI specificationに対する適合性の評価試験を行うために使用する適合性試験評価プログラム構成（Conformance Test Suite（CTS））の仕様をまとめたものである。BioAPIは３階層構造（BioAPI適合アプリケーション、BioAPI フレームワーク、BioAPI適合BSP(Biometric Service Provider)を持ち、本仕様書ではこれらの中でBioAPIフレームワークについての適合性試験評価プログラム構成をまとめたものである。本仕様を基に国際規格ISO24709-3 Infromation technology－Conformance Testing for the biometric application programming interface(BioAPI)－Part 3: Test assertions for BioAPI frameworks策定のための寄書を行う予定である。

４．２　適用範囲

本仕様書は、ＢｉｏＡＰＩ２．０フレームワークの適合性試験の基本仕様と詳細仕様から構成され、基本仕様には、CTSの機能の概要と適合性試験の対象となるBioAPI関数の試験方法および試験のシナリオをまとめる。詳細仕様では、基本仕様で定義された適合性試験の評価方法とシナリオを基として、適合性評価のための具体的な試験関数の定義をXMLにて記述する。

４．３　用語の意味

本文で使用する専門用語および略語は、ＩＳＯ／ＩＥＣ１９７８４－１：２００６およびＩＳＯ／ＩＥＣ２４７０９－１：２００７で定義されている。

（1）ＩＳＯ／ＩＥＣ１９７８４－１：２００６

a. 専門用語

1) BioAPI Compornent

2)BioAPI Function Provider (BFP)

3)BioAPI Unit

4)biometric data block (BDB)

5)biometric information record (BIR)

6)reference BIR

7)sample BIR

8)biometric sample

9)biometric template

10)intermediate biometric sample

11)processed biometric sample

12)raw biometric sample

13)reference template

14)biometric sensor

15)biometric service provider (BSP)

16)biometrics

17)callback

18)compornet registory

19)enrollment

20)False Match Rate (FMR)

21)handle

22)identify, identification

23)match

24)payload

25)score, scoring

26)security block

27)universal unique identifier (UUID)

28)verify, verification

b.略語

1) API - Application Programming Interface

2)BDB - Biometric Data Block

3)BFP - Biometric Function Provider

4)BIR - Biometric Infromation Record

5)BSP - Biometric Service Provider

6)CBEFF - Common Biometric Exchange Framework

7)FMR - False Match Rate
8)FPI - Function Provider Interface
9) ID – Identify/Identification/Identifier
10)SPI – Service Provider Interface
11) UUID – Universal Unique Identifier

（2）ＩＳＯ／ＩＥＣ　２４７０９－１：２００７

a. 専門用語

1)abstract test engine
2)base standard
3)BioAPI conformance test suite
4)BioAPI conformity statement
5)certification
6)conformance
7)conformance requirement
8)Comformitiy assessment
9)implementation under test
10)standard BioAPI compornent
11)standard BioAPI interface
12)test assertion
13)test method implementation
14)test case
15)test method implementation
16)test method specification
17)test puorpose
18)test report
19)test result code, test verdict
20)validation

c. 略語

1)BCS – BioAPI Conformity Statement
2)CTS – Conformance Test Suite
3)IUT – Implementation Under Test

４．４　基本仕様

本仕様書は、ＢｉｏＡＰＩ２．０フレームワークの適合性試験の基本仕様と詳細仕様から構成され、基本仕様には、CTSの機能の概要と適合性試験の対象となるBioAPI関数の試験方法および試験のシナリオをまとめる。詳細仕様では、基本仕様で定義された適合性試験の評価方法とシナリ

オを基として、適合性評価のための具体的な試験関数の定義をテストアサーションとしてXML言語にて記述する。

４．４．１　適合試験の方法論

国際規格ＩＳＯ／ＩＥＣ１９７８４－１およびＩＳＯ／ＩＥＣ２４７０９－１で定義されているＢｉｏＡＰＩ２．０フレームワークの適合性に関する定義および適合評価試験の方法及び手続きに準拠刷る方式を基本的に踏襲する方向ですすめる。

ＢｉｏＡＰＩ２．０の国際規格ＩＳＯ／ＩＥＣ１９７８４－１（ＢｉｏＡＰＩ２．０規格）に準拠するフレームワークの適合評価試験の方法論については、ＢｉｏＡＰＩ２．０規格書Ａｎｎｅｘ　AおよびＩＳＯ／ＩＥＣ２４７０９－１にて定義されている。以下に骨子を記載する。

（1）フレームワーク適合性評価試験の対象となるBioAPIインターフェイス関数は次のものである。

ａ）ＢｉｏＡＰＩ関数（４９関数）

-BioAPI_Init
- BioAPI_Terminate
- BioAPI_GetFrameworkInfo
- BioAPI_EnumBSPs
- BioAPI_BSPLoad
- BioAPI_BSPUnload
- BioAPI_BSPAttach
- BioAPI_BSPDetach
- BioAPI_QueryUnits
- BioAPI_EnumBFPs
- BioAPI_QueryBFPs
- BioAPI_ControlUnit
- BioAPI_FreeBIRHandle
- BioAPI_GetBIRFromHandle
- BioAPI_GetHeaderFromHandle
- BioAPI_EnableEvents
- BioAPI_SetGUICallbacks
- BioAPI_Capture
- BioAPI_CreateTemplate
- BioAPI_Process
- BioAPI_ProcessWithAuxBIR
- BioAPI_VerifyMatch
- BioAPI_IdentifyMatch
- BioAPI_Enroll
- BioAPI_Verify

- BioAPI_Identify
- BioAPI_Import
- BioAPI_PresetIdentifyPopulation
- BioAPI_DbOpen
- BioAPI_DbClose
- BioAPI_DbCreate
- BioAPI_DbDelete
- BioAPI_DbSetMarker
- BioAPI_DbFreeMarker
- BioAPI_DbStoreBIR
- BioAPI_DbGetBIR
- BioAPI_DbGetNextBIR
- BioAPI_DbDeleteBIR
- BioAPI_SetPowerMode
- BioAPI_SetIndicatorStatus
- BioAPI_GetIndicatorStatus
- BioAPI_CalibrateSensor
- BioAPI_Cancel
- BioAPI_Free
- BioAPI_Util_InstallBSP
- BioAPI_Util_InstallBFP

b）ＢｉｏＳＰＩ関数（３９関数）

- BioSPI_BSPLoad
- BioSPI_BSPUnload
- BioSPI_BSPAttach
- BioSPI_BSPDetach
- BioSPI_QueryUnits
- BioSPI_QueryBFPs
- BioSPI_ControlUnit
- BioSPI_FreeBIRHandle
- BioSPI_GetBIRFromHandle
- BioSPI_GetHeaderFromHandle
- BioSPI_EnableEvents
- BioSPI_SetGUICallbacks
- BioSPI_Capture
- BioSPI_CreateTemplate
- BioSPI_Process

- BioSPI_ProcessWithAuxBIR
- BioSPI_VerifyMatch
- BioSPI_IdentifyMatch
- BioSPI_Enroll
- BioSPI_Verify
- BioSPI_Identify
- BioSPI_Import
- BioSPI_PresetIdentifyPopulation
- BioSPI_DbOpen
- BioSPI_DbClose
- BioSPI_DbCreate
- BioSPI_DbDelete
- BioSPI_DbSetMarker
- BioSPI_DbFreeMarker
- BioSPI_DbStoreBIR
- BioSPI_DbGetBIR
- BioSPI_DbGetNextBIR
- BioSPI_DbDeleteBIR
- BioSPI_SetPowerMode
- BioSPI_SetIndicatorStatus
- BioSPI_GetIndicatorStatus
- BioSPI_CalibrateSensor
- BioSPI_Cancel
- BioSPI_Free

ｃ）アプリケーション・コールバック・インターフェイス

- BioAPI_EventHandler
- BioAPI_GUI_STATE_CALLBACK
- BioAPI_GUI_STREAMING_CALLBACK

ｄ）フレームワーク・コールバック・インターフェイス

- BioSPI_EventHandler
- BioSPI_GUI_STATE_CALLBACK
- BioSPI_GUI_STREAMING_CALLBACK
- BioSPI_BFP_ENUMERATION_HANDLER
- BioSPI_MEMORY_FREE_HANDLER

（2）ＩＳＯ／ＩＥＣ１９７８４－１での適合性に関する定義

Ａｎｎｅｘ　Ａにおいて、以下の要件が規定されている。

- ＢＳＰのロード/取り付け（アタッチ）

ＢＳＰとＢＦＰの管理
コンポーネント・レジストリーのメンテナンスと管理

- ＡＰＩ関数とＳＰＩ関数の間の透過形の関数呼び出しを行う

ＢＳＰからのイベント発生の通知処理と、これらのイベント通知をＢＳＰをロードしたアプリケーション内の（可能性としては、複数の）イベントハンドラーへ送信すること
ＢｉｏＡＰＩコンポーネントのインストールと取り外し（デインストール）に関連したＡＰＩ関数呼び出しを行うと共に、コンポーネント・レジストリーの関連する内容を更新すること
インストールしたＢＦＰに関するＢＳＰからの問い合わせをサポートすること
コンポーネント・レジストリの機能と処理が用意されること
ＡＰＩ関数とこれに関連するＳＰＩ関数との間の翻訳変換を実行すること
エラーコードに準拠すること
イベント通知とコールバックを処理すること
ＩＳＯ／ＩＥＣ１９７８４－１：２００６で定義された全てのオプションをサポートすること

（3）　ＩＳＯ／ＩＥＣ２４７０９－１：２００７での定義

A. 適合試験の対象となる実装体（Ｉｍｐｌｅｍｅｎｔａｔｉｏｎ　Ｕｎｄｅｒ　Ｔｅｓｔ（ＩＵＴ））の定義

- ＢｉｏＡＰＩ規格の要件定義に準拠するソフトウェアとハードウェア

B. 要件定義は以下の３種類に分類されるが、その各々について実装される場合は、ＢｉｏＡＰＩの定義に準拠しなければならない。

- 必須要件
- 条件付要件
- オプション要件

このために、ＢｉｏＡＰＩ適合性報告書（ＢｉｏＡＰＩ　Ｃｏｍｆｏｒｍａｎｃｅ　Ｓｔａｔｅｍｅｎｔ）が必須となる。この報告書の内容により適合評価試験の判定項目を選定し、その判定項目の要件がみたされているかどうかを検証する。ただし、BioAPIフレームワームの場合は、ＩＳＯ／ＩＥＣ１９７８４－１Ａｎｎｅｘ　Ａでのフレームワークに対する適合性の定義で全てのオプションをサポートすることが要求されているため、フレームワーク用ＢｉｏＡＰＩ適合性報告書は下記になると考えれる。
ＢｉｏＡＰＩ適合性報告書(BioAPI Comformance Statement) ―フレームワーク用―

表４－１　フレームワーク用ＢｉｏＡＰＩ適合性報告書

Vender Contact information	
Name	

Vender contact information:	
Street	
City	
State or province	
Zip code or postal code	
Country	
Telephone	
Biometric product:	
Name	
Serial number	
description	
BioAPI conformance class (one of the following):	
BioAPI application	
BioAPI framework (this alternative shall be selected)	
BSP	
Additional information	
UUID of the framework	
BioAPI specification version	
Product version	
Supported BDB formats (one or more format owner/formant type pairs)	
Supported biometric types (one or more)	
Maximum supported size for the payload	

（4）適合性試験ソフトウェアの構成

適合性試験ソフトウェア(Conformance Test Suite (CTS))の構成は下図のようになる。試験の準備段階において、グラフィックユーザーインターフェイスを利用して、適合評価試験の検査項目、BioAPI適合性記述文（BioAPI適合性報告書より作成されたXML記述文書）が用意され、適合試験検証内容定義が決定されシステムに入力される。このデータは、適合試験プロセス管理により、適合試験エンジンに渡され試験対象（フレームワーク）の試験を開始する。試験の結果はテストログとして出力され、適合試験レポート生成機により適合試験テスト報告書が生成される。

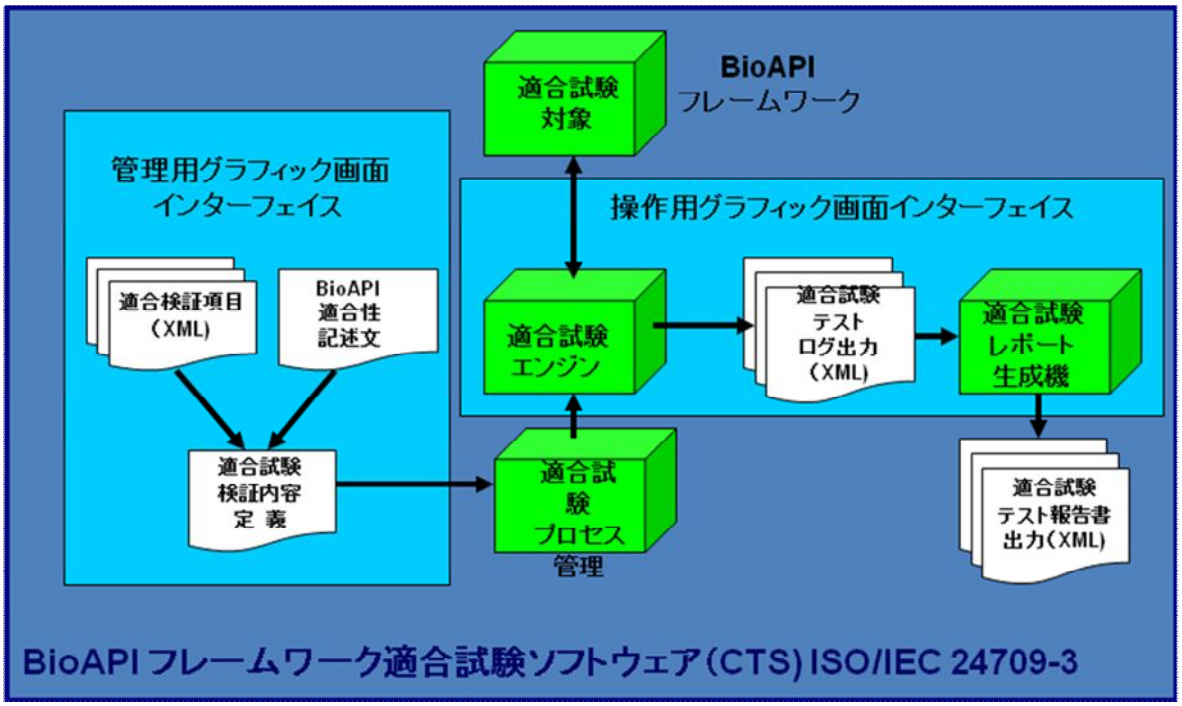


図４－１　適合性試験ソフトウェア(Conformance Test Suite (CTS))の構成

テストログとレポートの例を下記に記載する。

・テストログの例

```
<?xml version="2.0" encoding="UTF-8"?>
<conformance_test_log date_time="2007-01-20T12:06:45-05:00" standard="ISO/IEC 19784
BioAPI">
<TestingLaboratory>
<Name></Name>
<Address>
<Street> </Street>
<City>Tokyo</City>
<StateOrProvince>Tokyo</StateOrProvince>
<ZipOrPostalCode></ZipOrPostalCode>
<Country>JAPAN</Country>
</Address>
<Phone> </Phone>
<Email></Email>
<Url></Url>
</TestingLaboratory>
<Vendor>
```

<Name> </Name>

<Address>

<Street/>

<City/>

<StateOrProvince/>

<ZipOrPostalCode> </ZipOrPostalCode>

<Country>JAPN</Country>

</Address>

<Phone/>

<Email> </Email>

<Url> </Url>

</Vendor>

<Biometric_Product Name=" ">

<Description>

BioAPI2.0 Framework

</Description>

</Biometric_Product>

<CTS_ID> </CTS_ID>

<test_assertion>

<package_name> </package_name>

<assertion_name>BioAPI_Enroll_And_BioSPI_Enroll_Payload</assertion_name>

<description>

This assertion checks if calling BioAPI_Enroll with a payload returns

BioAPI_OK.

BioAPI_RETURN BioAPI BioSPI_Enroll

(BioAPI_HANDLE ModuleHandle,

BioAPI_BIR_PURPOSE Purpose,

const BioAPI_INPUT_BIR *StoredTemplate,

BioAPI_BIR_HANDLE_PTR NewTemplate,

const BioAPI_DATA *Payload,

sint32 Timeout

BioAPI_BIR_HANDLE_PTR AuditData);

This function captures biometric data from the attached device for the

purpose of creating a ProcessedBIR for the purpose of enrollment.

Section 2.5.3.6:

Parameters: Payload (input/optional) - a pointer to data that will be

wrapped inside the newly created template.
This parameter is ignored, if NULL.
Section 1.6:
The BSP may optionally allow the application to provide a 'payload' to wrap inside the new template.
Section 2.2.1.2:
#define BioAPI_PAYLOAD (0x00001000)
If set, indicates that the BSP supports payload carry (accepts payload during enroll/process and returns payroll upon successful verify).

In order to determine conformance with respect to the text above, the following steps are performed:
1) Load the BSP under test
2) Attach the BSP under test
3) Call BioAPI_Enroll to capture and enroll a BIR with payload set to a non-NULL value.
4) Check the return code, which is expected to be BioAPI_OK.
If any of the intermediate operations fail, an UNDECIDED conformity response is issued.
</description>
<input name="_BSPUuid" value=" "/>
<input name="_VersionMajor" value="2"/>
<input name="_VersionMinor" value="0"/>
<input name="_inserttimeout" value="15000"/>
<input name="_noSourcePresentSupported" value="true"/>
<input name="_sourcepresenttimeout" value="15000"/>
<input name="_capturetimeout" value="15000"/>
<input name="_supportPayload" value="true"/>
<input name="_payload" value="0123456789"/>
</test_assertion>
<inline_response>
<asserted_condition>
The input value of the parameter "Reserved" of the event handler is 0.
</asserted_condition>
<conformance>pass</conformance>
</inline_response>
<function dir="outgoing">

```
<name>BioAPI_ModuleEventHandler</name>
<input name="BSPUuid" value=""/>
<input name="BioAPINotifyCallbackCtx" value="*"/>
<input name="DeviceID" value="0"/>
<input name="Reserved" value="0"/>
<input name="EventType" value="1"/>
<return value="0"/>
</function>
<function dir="incoming">
<name>BioAPI_BSPLoad</name>
<input name="Reserved" value="0"/>
<input name="BSPUuid" value=""/>
<input name="BioAPINotifyCallback" value="*"/>
<input name="BioAPINotifyCallbackCtx" value="*"/>
<return value="0"/>
</function>
<inline_response>
<asserted_condition>
The function BioAPI_BSPLoad has returned BioAPI_OK.
</asserted_condition>
<conformance>pass</conformance>
</inline_response>
<inline_response>
<asserted_condition>
The BioAPI_NOTIFY_INSERT event notification has been received within
the specified maximum duration
</asserted_condition>
<conformance>pass</conformance>
</inline_response>
<function dir="incoming">
<name>BioAPI_BSPAttach</name>
<input name="BSPUuid" value=" "/>
<input name="VersionMajor" value="2"/>
<input name="VersionMinor" value="0"/>
<input name="DeviceID" value="0"/>
<input name="Reserved1" value="0"/>
<input name="Reserved2" value="0"/>
<input name="ModuleHandle" value="1"/>
```

<input name="Reserved3" value="0"/>
<input name="Reserved4" value="0"/>
<input name="Reserved5" value="0"/>
<input name="Reserved6" value="0"/>
<return value="0"/>
</function>
<inline_response>
<asserted_condition>
The function BioSPI_BSPAttach has returned BioAPI_OK.
</asserted_condition>
<conformance>pass</conformance>
</inline_response>
<inline_response>
<asserted_condition>
Either the framework under test does not transfer the
BioAPI_NOTIFY_SOURCE_PRESENT event notification, or the event notification has been sent
within the specified maximum duration.
</asserted_condition>
<conformance>pass</conformance>
</inline_response>
<function dir="incoming">
<name>BioAPI_Enroll</name>
<input name="ModuleHandle" value="1"/>
<input name="Purpose" value="4"/>
<input name="StoredTemplate_Form" value=""/>
<input name="StoredTemplate_DbHandle" value=""/>
<input name="StoredTemplate_KeyValue" value=""/>
<input name="StoredTemplate_BIRHandle" value=""/>
<input name="StoredTemplate_Length" value=""/>
<input name="StoredTemplate_HeaderVersion" value=""/>
<input name="StoredTemplate_ProcessedLevel" value=""/>
<input name="StoredTemplate_Encrypted" value=""/>
<input name="StoredTemplate_Signed" value=""/>
<input name="StoredTemplate_FormatOwner" value=""/>
<input name="StoredTemplate_FormatId" value=""/>
<input name="StoredTemplate_Quality" value=""/>
<input name="StoredTemplate_Purpose" value=""/>
<input name="StoredTemplate_FactorMultiple" value=""/>

```
<input name="StoredTemplate_FactorFacialFeatures" value=""/>
<input name="StoredTemplate_FactorVoice" value=""/>
<input name="StoredTemplate_FactorFingerprint" value=""/>
<input name="StoredTemplate_FactorIris" value=""/>
<input name="StoredTemplate_FactorRetina" value=""/>
<input name="StoredTemplate_FactorHandGeometry" value=""/>
<input name="StoredTemplate_FactorSignatureDynamics" value=""/>
<input name="StoredTemplate_FactorKeystrokeDynamics" value=""/>
<input name="StoredTemplate_FactorLipMovement" value=""/>
<input name="StoredTemplate_FactorThermalFaceImage" value=""/>
<input name="StoredTemplate_FactorThermalHandImage" value=""/>
<input name="StoredTemplate_FactorGait" value=""/>
<input name="StoredTemplate_FactorPassword" value=""/>
<input name="StoredTemplate_BiometricData" value=""/>
<input name="StoredTemplate_Signature" value=""/>
<input name="no_NewTemplate" value=""/>
<input name="Payload" value="0123456789"/>
<input name="Timeout" value="15000"/>
<input name="no_AuditData" value=""/>
<output name="NewTemplate" value="2"/>
<output name="AuditData" value="-2"/>
<return value="0"/>
</function>
<inline_response>
<asserted_condition>
The function BioAPI_Enroll has returned BioAPI_OK
</asserted_condition>
<conformance>pass</conformance>
</inline_response>
<function dir="incoming">
<name>BioAPI_BSPDetach</name>
<input name="BSPHandle" value="1"/>
<return value="0"/>
</function>
<inline_response>
<asserted_condition>
The function BioAPI_BSPDetach has returned BioAPI_OK.
</asserted_condition>
```

```
<conformance>pass</conformance>
</inline_response>
<function dir="incoming">
<name>BioAPI_BSPUnload</name>
<input name="Reserved" value="0"/>
<input name="BSPUuid" value=" "/>
<input name="BioAPINotifyCallback" value="*"/>
<input name="BioAPINotifyCallbackCtx" value="*"/>
<return value="0"/>
</function>
<inline_response>
<asserted_condition>
The function BioAPI_BSPUnload has returned BioAPI_OK.
</asserted_condition>
<conformance>pass</conformance>
</inline_response>
</conformance_test_log>
```

表４－２　適合試験テスト報告書（Conformance Report)の例

Report ID	
Issue Date	
Date of Test	
CTS ID	
Testing Laboratory Vendor	
Biometric Product	
Name	
Serial No.	
Description	
Assertion	
Name	
Package	
Description	
Conformance	Pass/Fail

Report Authorization

Name **Title** **Signature**

４．４．２　適合試験の方法

適合試験の方法については、ＩＳＯ／ＩＥＣ２４７０９－１で以下の要件定義がされている。

- 実装体をブラックボックスとして試験を行う。
- 適合試験ソフトウェア（Ｃｏｎｆｏｒｍａｎｃｅ　Ｔｅｓｔ　ｓｕｉｔｅ（ＣＴＳ））を用いる
- 反証試験方法論を用いる。（”ＰＡＳＳ”／”ＦＡＩＬ”判定を行う）従って、非適合性の判定のみが評価できる。
- 試験の準備、実行、結果についての詳細な文書化を行うこと
  XLM表記によるテストログとテスト結果報告書の作成が必要である。
- ＢｉｏＡＰＩの適合性試験モデル（下図参照）に従うこと
- フレームワークでは、フレームワークテスト用アプリケーションとフレームワークテスト用ＢＳＰが必要である。

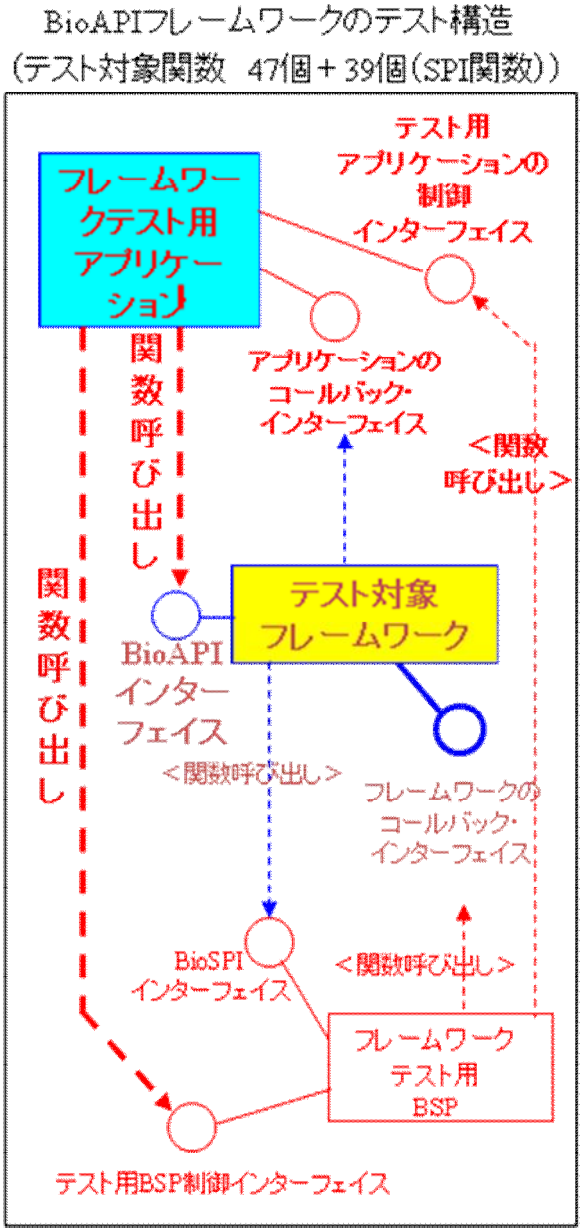


図４－３　ＢｉｏＡＰＩ２．０準拠フレームワームの適合評価試験モデル

４．４．３　適合性の判定方法

適合性の判定方法については、ＩＳＯ／ＩＥＣ２４７０９－１：２００７の9節にＢｉｏＡＰＩ準拠のフレームワークが備えているＢｉｏＡＰＩ関数、ＢｉｏＳＰＩ関数各々についての定義を用いる。

４．５　詳細仕様

４．５．１　はじめに

上記基本仕様に基づき、適合性試験のアサーションをXML表記により下記に定義する。

４．５．２　適用範囲

本詳細仕様は、ＩＳＯ／ＩＥＣ１９７８４－１：２００６に準拠したと宣言されたフレームワーク製品についての適合評価試験を実行するソフトウェアに実装されるテストアサーションを定義する。本詳細仕様内で定義されるテストアサーションは、ＩＳＯ／ＩＥＣ２４７０９－１：２００７にて定義された

アサーション言語（XML）および適合評価試験の方法論、手続きを基本として、基本仕様でまとめられた要求仕様を満たすものである。

４．５．３　試験詳細（テストアサーション）

(1)　　一般事項

本テストアサーションは、共通アクティビティと６４種類の個別アサーションから構成される。

(2)　　共通アクティビティ

以後の付属節にて定義されている試験詳細定義（テストアサーション）について共通に使用できるテストアサーションを添付にまとめる。

５　OSS Nokalva BioFoundry社製 BioAPI2.0準拠フレームワークを活用したテスト

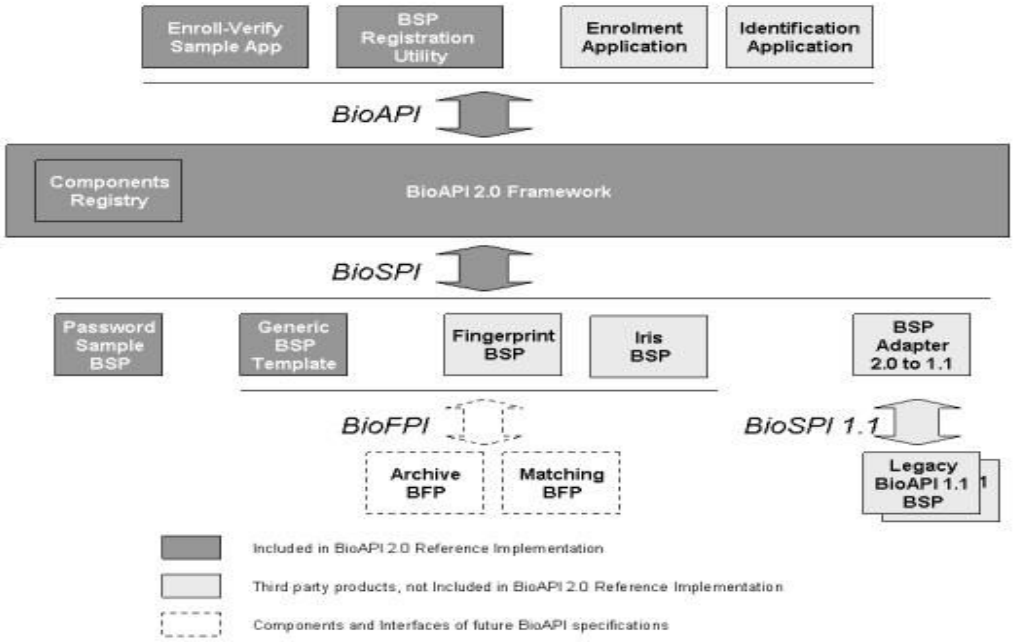
５．１　はじめに

ＢｉｏＡＰＩ２．０準拠のフレームワークの適合性を検証する試験方法を検討するため、全世界において公開されているフレームワークの機能、性能を調査、解析する必要がある。

現在、公開されているフレームワークは、米国ＯＳＳ　Ｎｏｋａｌｖａ社（以下ＯＳＳ社と省略する）のフレームワークのみである。このため、ＯＳＳ社からの許諾を得てフレームワークの機能、性能の調査を開始した。以下に今年度実施した内容と、今後の予定をまとめる。なお、ＯＳＳ社の詳細に関しては、ウェブサイト（http://www.oss.com/）を参照していただきたいが、以下に簡単にこの会社のバイオメトリクス分野での活動を記載する。

ＯＳＳ社は１５年以上にわたりネットワーク・プロトコル、ＡＳＮ．１、XML、バイオメトリクスの分野において、基礎技術の開発を推進する企業として活動している。特にバイオメトリクスの分野においては、２００３年から始まったＩＳＯ／ＩＥＣ１９７８４ＢｉｏＡＰＩの規格開発活動において、米国からの寄書、基礎となる研究活動において中心的な役割を担ってきている。この中でＯＳＳ社は世界に先駆けて、ＢｉｏＡＰＩ２．０準拠のフレームワークを開発し、販売を行っている。ＯＳＳ社は、ＳＣ３７WG２において活発な活動を行っており、現在ＩＳＯ／ＩＥＣ２４７０４Ｂｉｏｍｅｔｒｉｃ　Ｉｎｔｅｒｎｅｔｗｏｒｋｉｎｇ、およびＩＳＯ／ＩＥＣ１９７８４－１Ａｍｄ４、ＩＳＯ／ＩＥＣ１９７８５－４において大きな貢献を行っている。

ＯＳＳのフレームワークについては、下図に示すように種々のインターフェイス機能を用意している。



２月始めにＪＡＩＳＡとの契約が成立し、ＯＳＳ社から入手したフレームワークの内容と今年度実施したテストの結果を以下にまとめる。

５．２　ソフトウェア構成

入手したＢｉｏＡＰＩ２．０準拠フレームワークに関するソフトウェアは以下の表５．１に示すものとなっている。

表5－1　OSS社　BioAPI2．0フレームワーク関連ファイル構成

種類	ファイル名
インストーラー	BioFoundrySDK_Standard_Setup.exe
ライセンス	biofoundry-sdk.lic
ReadMe （docフォルダ）	OSSlogo.gif ReadMe.doc ReadMe.html SDK overview top.jpg SDK overview.jpg
サンプルBSPプログラム （SampleBSPsフォルダ）	PwdDynBSP.dll
ドキュメント （SDK_COM¥docフォルダ）	Index.html
サンプルプログラム （SDK_CPP¥Binフォルダ）	EnrollVerify.exe RegBSP20.exe
サンプルプログラム共通ソースファイル （SDK_CPP¥Commonフォルダ）	bioapi_errdef.h CErrors.cpp CErrors.h CList.cpp CList.h CLock.cpp CLock.hpp CRWSyncObject.hpp CSchema.cpp CSchema.h environ.h product_id.h sys_w32.cpp sys_w32.hpp util.cpp util.hpp
サンプルプログラム用ドキュメント （SDK_CPP¥Docフォルダ）	Building.html Content.html Index.html Install.html OSSlogo.gif Overview.gif Overview.html

	RunningSamples.html
サンプルプログラム用インクルードファイル （SDK_CPP¥Includeフォルダ）	bioapi.h bioapi_api.h bioapi_err.h bioapi_porttype.h bioapi_reg.h bioapi_spi.h bioapi_type.h biospi.h oss_hdr.h
一括ビルド用ソリューションファイル （SDK_CPP¥Samplesフォルダ）	BuildAll.sln
登録照合サンプルプログラム用ソース （SDK_CPP¥Samples¥EnrollVerifyフォルダ）	EnrollVerify.cpp EnrollVerify.h EnrollVerify.rc EnrollVerify.sln EnrollVerify.vcproj EnrollVerifyBSP.cpp EnrollVerifyBSP.h EnrollVerifyDlg.cpp EnrollVerifyDlg.h resource.h StdAfx.cpp StdAfx.h
登録照合サンプルプログラム用リソース （SDK_CPP¥Samples¥EnrollVerify¥Resフォルダ）	EnrollVerify.ico EnrollVerify.rc2
レジストリ登録サンプルプログラム用ソース （SDK_CPP¥Samples¥RegBSP20フォルダ）	RegBSP20.cpp RegBSP20.h RegBSP20.rc RegBSP20.sln RegBSP20.vcproj RegBSP20Dlg.cpp RegBSP20Dlg.h resource.h stdafx.cpp stdafx.h
レジストリ登録サンプルプログラム用リソース （SDK_CPP¥Samples¥RegBSP20¥Resフォルダ）	RegBSP20.ico RegBSP20.rc2
フレームワーク本体及びインストーラー （Shared¥Framework¥Binフォルダ）	bioapi20.dll crds_sysreg.dll Framework.inf framework_install.exe framework_uninstall.exe
Win9x用フレームワーク本体及びインストーラー （Shared¥Framework¥Bin¥Win98フォルダ）	bioapi20.dll crds_sysreg.dll Framework.inf framework_install.exe

	framework_uninstall.exe
フレームワーク用ドキュメントファイル （Shared¥Framework¥Docフォルダ）	Content.html Install.html OSSlogo.gif ReadMe.html
フレームワーク用ライブラリファイル （Shared¥Framework¥Libフォルダ）	bioapi20.lib
エラー表示及びレジストリ表示ユーティリティー （Utilsフォルダ）	BioError.exe BioRegistry.exe

フレームワーク本体のモジュールは、ｂｉｏａｐｉ２０．ｄｌｌであり、動作チェック用のプログラムとして、サンプルＢＳＰプログラムＰｗｄＤｙｎＢＳＰ．ｄｌｌ、ＥｎｒｏｌｌＶｅｒｉｆｙ．ｅｘｅ及びＲｅｇＢＳＰ２０．ｅｘｅが用意されている。ＯＳＳ　Ｎｏｋａｌｖａ社は、バイオメトリクスのセンサーを持たないため、サンプルＢＳＰとしてパスワード認証のＢＳＰをサンプルとして添付してあるのみであった。

５．３　動作テスト

今年度は、ＯＳＳ社が推奨する方式での動作テストを行った。以下にその内容と結果を記載する。

（テスト環境）

- 使用ＰＣ
  - ＨＰ　ｘｗ８４００／ＣＴ　Ｗｏｒｋｓｔａｔｉｏｎ
    - ＣＰＵ：　Ｉｎｔｅｌ　Ｘｅｏｎ　５１６０　３．０ＧＨｚ
    - メモリー：４ＧＢ
    - ＨＤ：　２５０ＧＢ
  - Ｄｅｌｌ　Ｌａｔｉｔｕｅ　Ｄ５２０
    - ＣＰＵ：Ｉｎｔｅｌ　Ｃｏｒｅ２　Ｄｕｏ　Ｔ７２００　２ＧＨｚ
    - メモリー：１ＧＢ
    - ＨＤ：　８０ＧＢ
- 使用ＯＳ：　Ｗｉｎｄｏｗｓ　Ｖｉｓｔａ　（英語版、日本語版）
- プログラム開発用ソフト：　ＭＳ　Ｖｉｓｕａｌ　Ｓｔｕｄｉｏ２００５

（マイクロソフト製）

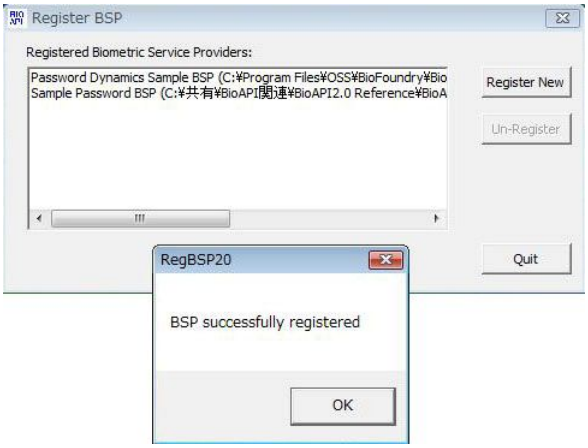
- テストソフトウェア：　ＢｉｏＡＰＩ２．０準拠フレームワーク

（ＯＳＳＮｏｋａｌｖａ製）
- 使用装置：　ＦＰ－ＰＬＵＳ指紋収集装置（セキュアデザイン株式会社製）

５．３．１　　ＯＳＳ推奨の動作テスト

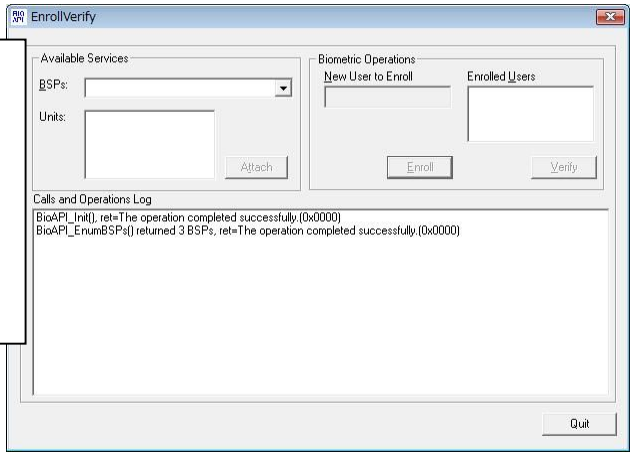
0SS がフレームワークテスト用として提供している　ＲｅｇＢＳＰ２．０．ｅｘｅおよびＥｎｒｏｌｌＶｅｒｉｆｙ．ｅｘｅを使用し、
これにセキュアデザイン（株）の指紋収集装置ＦＰ－ＰＬＵＳ、テンプレート生成モジュール及び指紋照合モジュールをＢＳＰとして組み込み動作テストを行った。
動作テストの流れを以下の図５－１に示す
。

操作画面（(1)



操作画面(2)

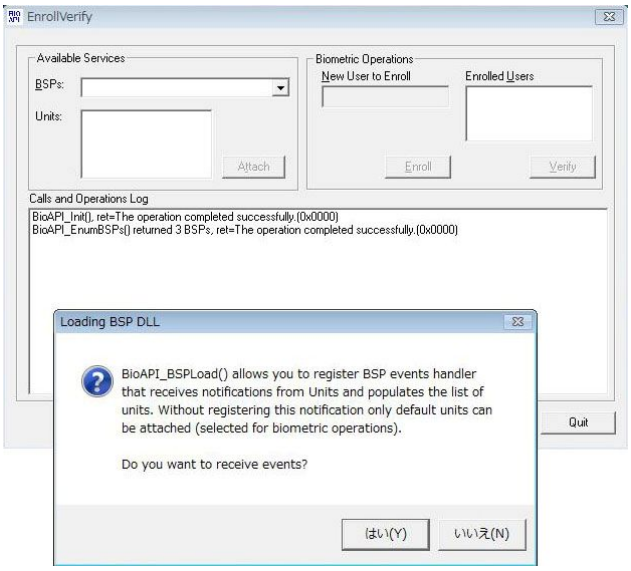
(BioAPI使用関数)
BioAPI_Init()
BioAPI_Free()
BioAPI_EnumBSP()



操作画面(3)

BSP選択

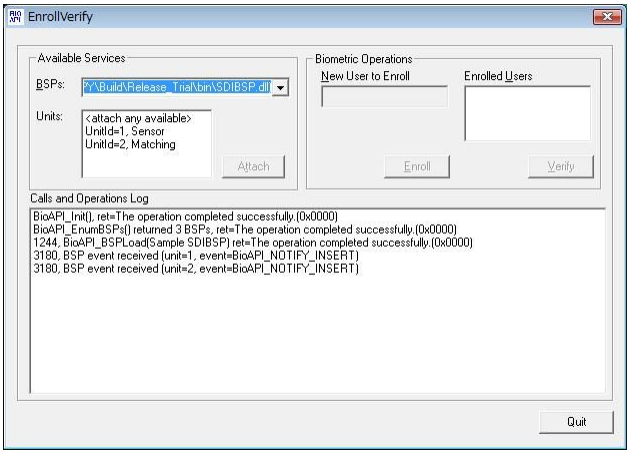
(BioAPI使用関数)
BioAPI_Init()
BioAPI_EnumBSP()


EnrollVerify
Available Services
BSPs:
Units:
Attach
Biometric Operations
New User to Enroll
Enrolled Users
Enroll
Verify
Calls and Operations Log
BioAPI_Init(), ret=The operation completed successfully.(0x0000)
BioAPI_EnumBSPs() returned 3 BSPs, ret=The operation completed successfully.(0x0000)
Loading BSP DLL
BioAPI_BSPLoad() allows you to register BSP events handler that receives notifications from Units and populates the list of units. Without registering this notification only default units can be attached (selected for biometric operations).
Do you want to receive events?
はい(Y)
いいえ(N)
Quit

操作画面(4)

BSPロード

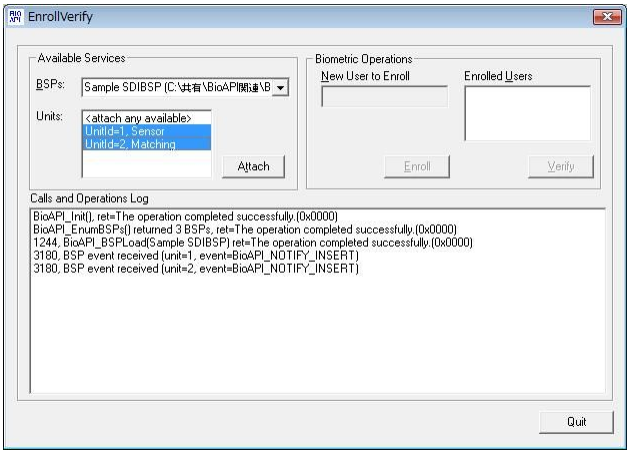
<BioAPI使用関数>
BioAPI_BSPLoad()



操作画面(5)

ユニット選択

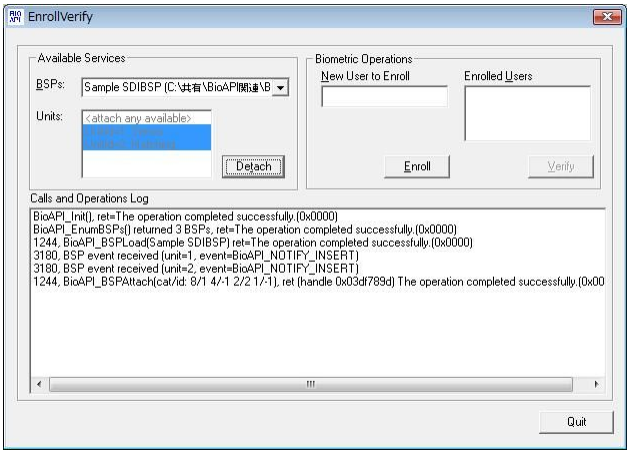
<BioAPI使用関数>
BioAPI_BSPLoad()



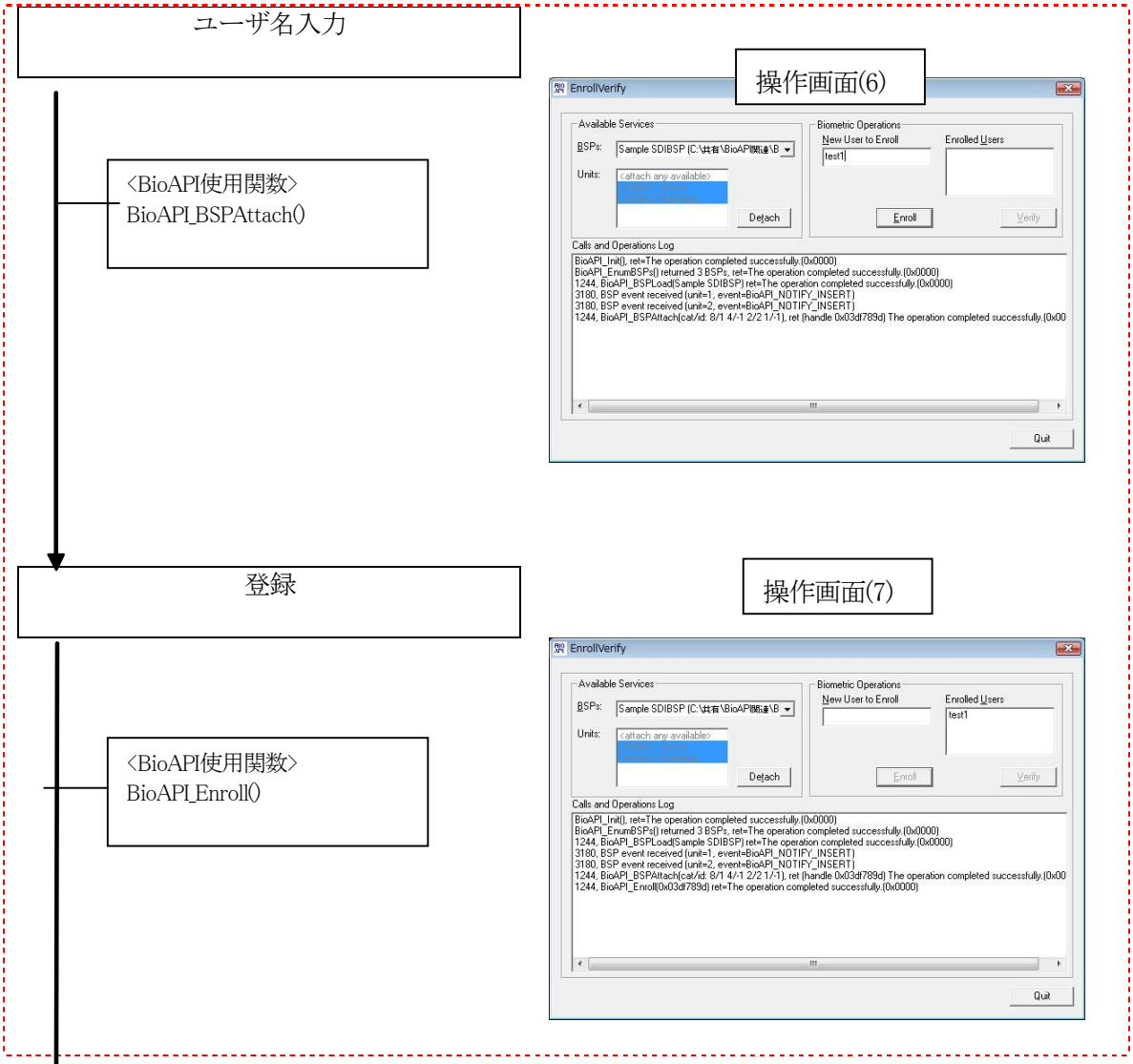
操作画面(6)

ユニットアタッチ

<BioAPI使用関数>
BioAPI_BSPAttach()



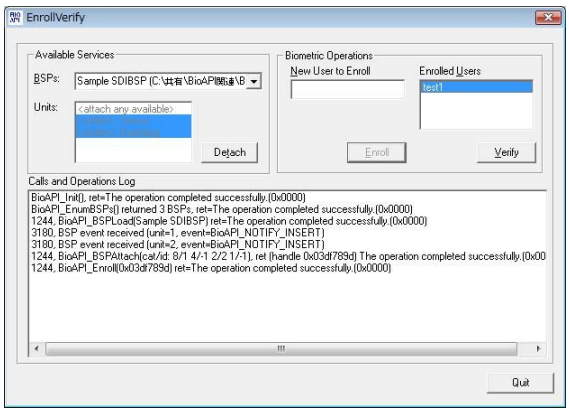
登録作業


ユーザ名入力
操作画面(6)
EnrollVerify
Available Services
BSPs:
Sample SDIBSP (C:\共有\BioAPI関連\B
Units:
<attach any available>
Detach
Biometric Operations
New User to Enroll
test1
Enrolled Users
Enroll
Verify
Calls and Operations Log
BioAPI_Init(), ret=The operation completed successfully.(0x0000)
BioAPI_EnumBSPs() returned 3 BSPs, ret=The operation completed successfully.(0x0000)
1244, BioAPI_BSPLoad(Sample SDIBSP) ret=The operation completed successfully.(0x0000)
3180, BSP event received (unit=1, event=BioAPI_NOTIFY_INSERT)
3180, BSP event received (unit=2, event=BioAPI_NOTIFY_INSERT)
1244, BioAPI_BSPAttach(cat/id: 8/1 4/-1 2/2 1/-1), ret (handle 0x03df789d) The operation completed successfully.(0x00
Quit
〈BioAPI使用関数〉
BioAPI_BSPAttach()
登録
操作画面(7)
EnrollVerify
Available Services
BSPs:
Sample SDIBSP (C:\共有\BioAPI関連\B
Units:
<attach any available>
Detach
Biometric Operations
New User to Enroll
Enrolled Users
test1
Enroll
Verify
Calls and Operations Log
BioAPI_Init(), ret=The operation completed successfully.(0x0000)
BioAPI_EnumBSPs() returned 3 BSPs, ret=The operation completed successfully.(0x0000)
1244, BioAPI_BSPLoad(Sample SDIBSP) ret=The operation completed successfully.(0x0000)
3180, BSP event received (unit=1, event=BioAPI_NOTIFY_INSERT)
3180, BSP event received (unit=2, event=BioAPI_NOTIFY_INSERT)
1244, BioAPI_BSPAttach(cat/id: 8/1 4/-1 2/2 1/-1), ret (handle 0x03df789d) The operation completed successfully.(0x00
1244, BioAPI_Enroll(0x03df789d) ret=The operation completed successfully.(0x0000)
Quit
〈BioAPI使用関数〉
BioAPI_Enroll()

照合作業

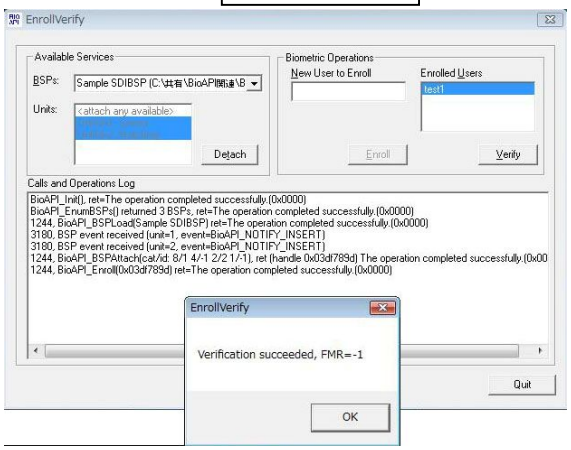
操作画面(8)

ユーザ名選択



照合

操作画面(9)



〈BioAPI使用関数〉
BioAPI_Verify()

終了

図５－１　動作テストの流れ

〈BioAPI使用関数〉
BioAPI_BSPDetach()
BioAPI_BSPUnload()
BioAPI_Terminate

５．３．２　セキュアデザイン研究所の指紋認証装置を利用した動作テスト
(FP-PLUS指紋収集装置の説明)

ＦＰ－ＰＬＵＳ指紋収集装置の仕様とフレームワークの動作確認のために作成したＦＰ－ＰＬＵＳ用簡易型ＢＳＰの内容を記載する。動作テストの流れは５．３．１節の５．１図動作テストの流れに示す手順でフレームワークを動作させ実行した。

（1）　ＦＰ－ＰＬＵＳ指紋読取ユニットの概要

指紋読取ユニットFP－PLUS／MBは、コンピュータネットワークセキュリティやデータベースセキュリティ及び個人認証を必要とするシステムに対し、指紋照合技術を利用したセキュリティ機能を提供する装置である。指紋読取ユニットFP－PLUS／MBは上位装置に接続し、上位装置の指示により、指紋の読み取りとデータの保存を実行することができる。

（2）　指紋読取ユニットの説明

ａ）構成品目

本製品は、下記に示す品目で構成される。

指紋読取ユニット本体　　１台

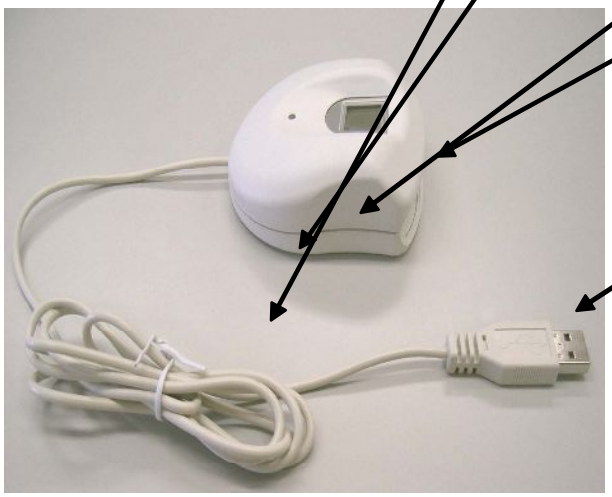
FP－PLUS／MB　環境設定用CD－ROM　　１枚

ｂ）設置条件

- 設置に適しているのは次のような場所である。
  - 直射日光の当たらない屋内
  - 温度 0℃～40℃
  - 湿度 20%～90%RH（但し結露なきこと）
  - 平坦で落下の恐れがない場所
- 次のような場所には設置しないこと。本機の故障や破損、誤動作が発生する場合がある。

・直射日光が当たる場所　　・暖房機の近く
・薬品や液体の近く　　・埃が多い場所

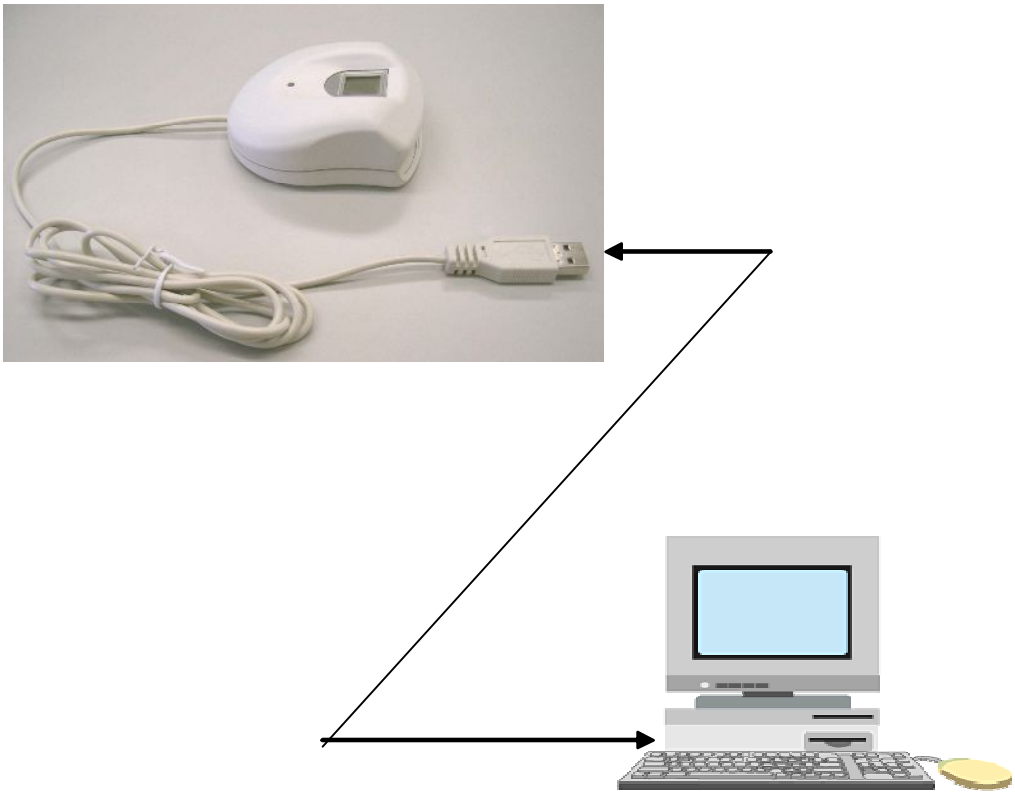
ｃ） 各部の名称と機能



	点灯	点滅
緑	電源ON	
オレンジ	指紋画像読み取り中	指紋画像読み取り待ち

ｄ） パソコンへの接続方法

本装置のUSBコネクタをパソコンのUSBコネクタ部に接続する。



注記）

本装置は、USBのバス電源を使用している。本装置の最大消費電流は250mAであり、複数のUSBデバイスを同時に接続している場合やUSBポートの電力量が小さい場合は、パソコン側の電源供給が不足することがある。また、電力量が足りている場合でも、パソコンとユニットの相性によりパソコンの電源供給が不足することがある。このような場合には、セルフパワー型（ハブ専用の電源を持つタイプ）USBハブで中継すると使用できる場合がある。

・装置の取り扱い

（１）本装置は精密機械であり、強い衝撃を与えないよう注意すること。

（２）センサ部をクリップやボールペンの先などの固い物でひっかかないこと。

（３）装置を落下させてしまった場合は、センサ部に破損がないことを必ず確認すること。

	センサ部に損傷のある状態で使用すると、怪我をする恐れや、装置の性能が出ない場合がある。

（４）センサ部の表面が糸くずや埃等で汚れている場合は、乾いた柔らかい布で清掃すること。詳細は、ｆ）項参照下のこと。

ｅ） 指の正しい置き方

（１）待機状態

LEDは緑色に点灯している。

（2）読み取り動作開始の表示

指紋を読み取る準備ができると、LEDはオレンジに点滅する。

（3）指の置き方と置く位置

指をセンサ部に乗せて、「カチッ」という音がするまで押すと、LEDにオレンジが点灯し、指紋画像を読み取る。LEDが緑色に点灯するまで指を離さないこと。指を置く位置については、図5-2を参照のこと。

注記）指を離すタイミングが早過ぎる場合や、指示に反して指を押し続けている場合は、指紋画像の読み取りができないことがある。

センサ部の表面が糸くずや埃等で汚れている場合は、読み取った指紋画像が図5-3のように暗くなることがある。装置の故障ではありませんが、指紋が正しく処理されない場合があります。ｆ）項に従い、センサ部を清掃してから再度試行すること。

指はセンサ部の中央部分に指紋の中心部分を乗せること。

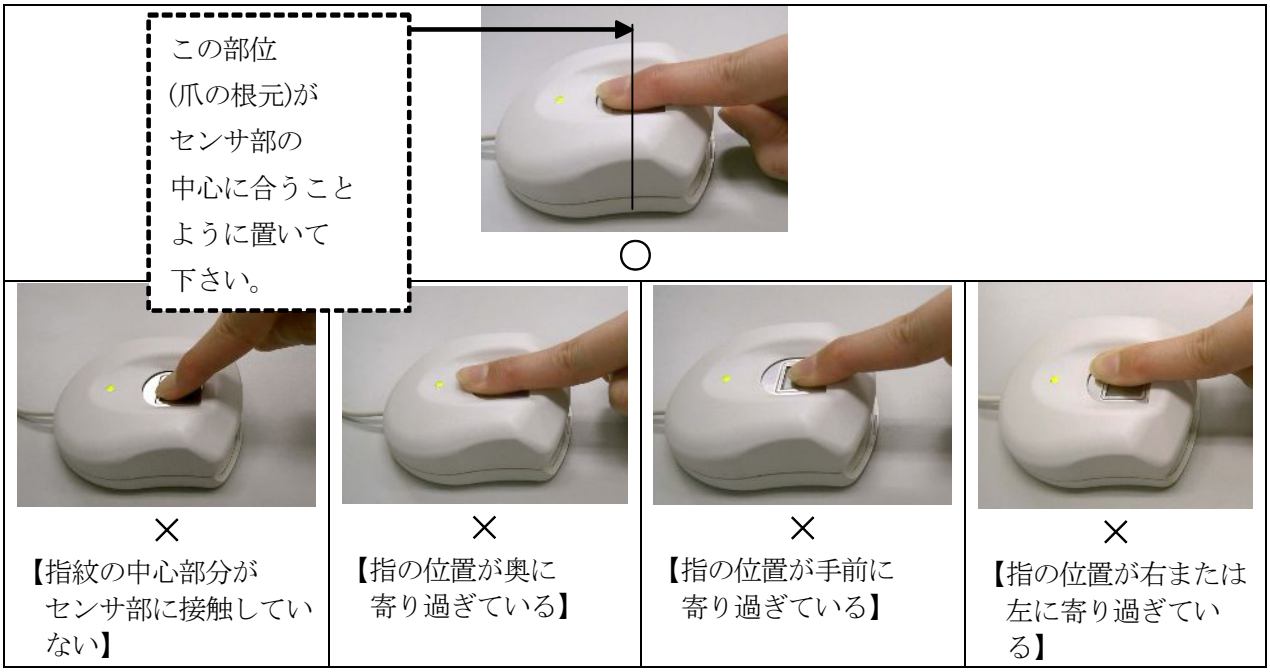


図５－２　指の置き方

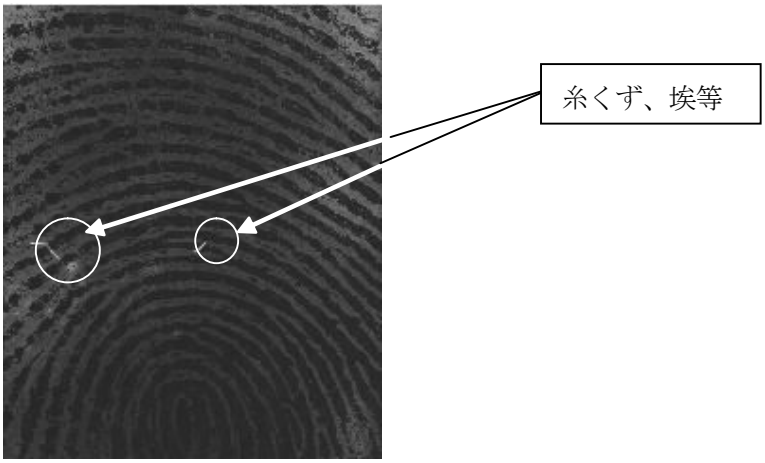


図５－３　指紋画像が暗くなった時の画像

ｆ）　日常のお手入れ

・センサ部は、汚れると誤作動が発生する場合があるので、適時乾いた柔らかい布で清掃すること。

・頑固な汚れには柔らかい布に水を含ませ、よく絞ってから拭くこと。

・本体とセンサ部の清掃に、有機溶剤や洗剤等は故障の原因となりますので使用しないこと。

ｇ）　装置仕様

一般規格

電源	USBコネクタより供給
消費電流	最大250mA
消費電力	最大1.25W
環境条件	使用温度　0℃～40℃ 保存温度　－25℃～60℃ 湿度　　20%～90%RH（結露なきこと）
適合規格	VCCI　クラスB　準拠
USB規格	USB Ver. 1.1 Full Speed 12Mbps対応
外形寸法	68mm(W)×82mm(D)×34 mm(H)
質量	約90g
静電耐圧	15kV以上

機能仕様

読み取り方式	散乱光蓄積電圧方式
読み取り領域	10mm(H)×12 mm(V)
読み取り密度	500dpi
読み取り階調	256階調

ｈ）その他

本製品を使用してトラブルが発生したり、故障と思われる症状が起きたら、まず、この章を参考にチェックすること。

もし、この項目にないような症状が起こったり、「本章で述べる対策」を行っても症状が消えない場合は、販売店へ問い合わせること。

その際は、ご使用の機器名称、故障時の詳しい状況、現在の状況を連絡すること。

i）最初に確認すること

・PCに正しく接続されているか確認すること。

j） PCが指紋読取ユニットを認識しない場合

原因	対処法
（1）ドライバが正しくインストールされていない	「FP－PLUS／MB　インストールマニュアル」を参照し、再度ドライバをインストールする。
（2）通信エラーが発生している	USBケーブルを抜き差しし、再接続する。
（3）パソコンからの供給電力不足	他のUSBデバイス装置を取り外した後、再度試行する。 ノートパソコンの機種によっては省電力モードを解除することにより動作可能となる。 e）項 パソコンへの接続方法の「メモ」を参照。

k）何度もドライバインストールを要求される場合

原因	対処法
（1）USBポート毎にドライバをインストールしていない	USBポート毎に、ドライバのインストールが必要である。 「FP－PLUS／MB　インストールマニュアル」を参照し、ドライバをインストールすること。 下記の「注記」を参照すること。

l） OSがストールする場合

原因	対処法
（1）他のUSBデバイス装置の抜き差しを行った	PCに過負荷がかかる事になり、OSが不安定になる場合がある。 FP－PLUS／MBで指紋を読み取っている最中は、PCに過負荷を与える処理を行わないこと。
（2）パソコンからの供給電力不足	他のUSBデバイス装置を取り外した後、再試行すること。 ノートパソコンの機種によっては省電力モードを解除すること。 e）項パソコンへの接続方法の「メモ」参照のこと。

注記）複数のUSBコネクタを持つパソコン等において、指紋読取ユニットの接続先を別のUSBコネクタに変更した場合は、ドライバのインストールが完了しているパソコンであっても、再度ドライバのインストールを要求されることがある。
この場合、最初にインストールした時と同様の手順で再度インストールを行うこと。

m）　登録または照合でエラーとなる場合

注記）指をセンサ部に乗せる時は、「カチッ」という音がするまで指を押すこと。

原因		対処法
（1）指の置き方が正しくない		e）項に従い、指を正しく置き直すこと。
（2）正しい操作を行っていない		e）項に従い、指をセンサ部に乗せて、「カチッ」という音がするまで押すこと。
（3）指の質	乾燥指の場合	異なる指に変更し、登録／照合を行うこと。 指に息を吹きかけたりして、適度な湿潤状態にし、再度、センサ部に指を置くこと。
	湿潤指の場合	異なる指に変更し、登録／照合を行うこと。 ハンカチ等の布で指を拭いた後、再試行すること。
	指紋の特徴点が少ない	異なる指に変更し、登録を行うこと。
	怪我、肌荒れがある	異なる指に変更し、登録を試みること。 複数指を登録してある場合、登録済みの別の指で照合を行うこと。
（4）センサ部の汚れ付着		５．３．２項を参考にセンサ部の清掃を行うこと。

（3）ＢＳＰ関数とＦＰ－ＰＬＵＳオリジナル関数のとの対応

ＢｉｏＳＰＩ関数とＦＰ－ＰＬＵＳオリジナル関数の対応関係は以下のようになる。この対応関係を利用してＦＰ－ＰＬＵＳをＢＳＰとしてフレームワークに登録し動作テストを行った。テストは、図５－４　動作テストの流れに示す手順でプログラムを動作させて実行した。

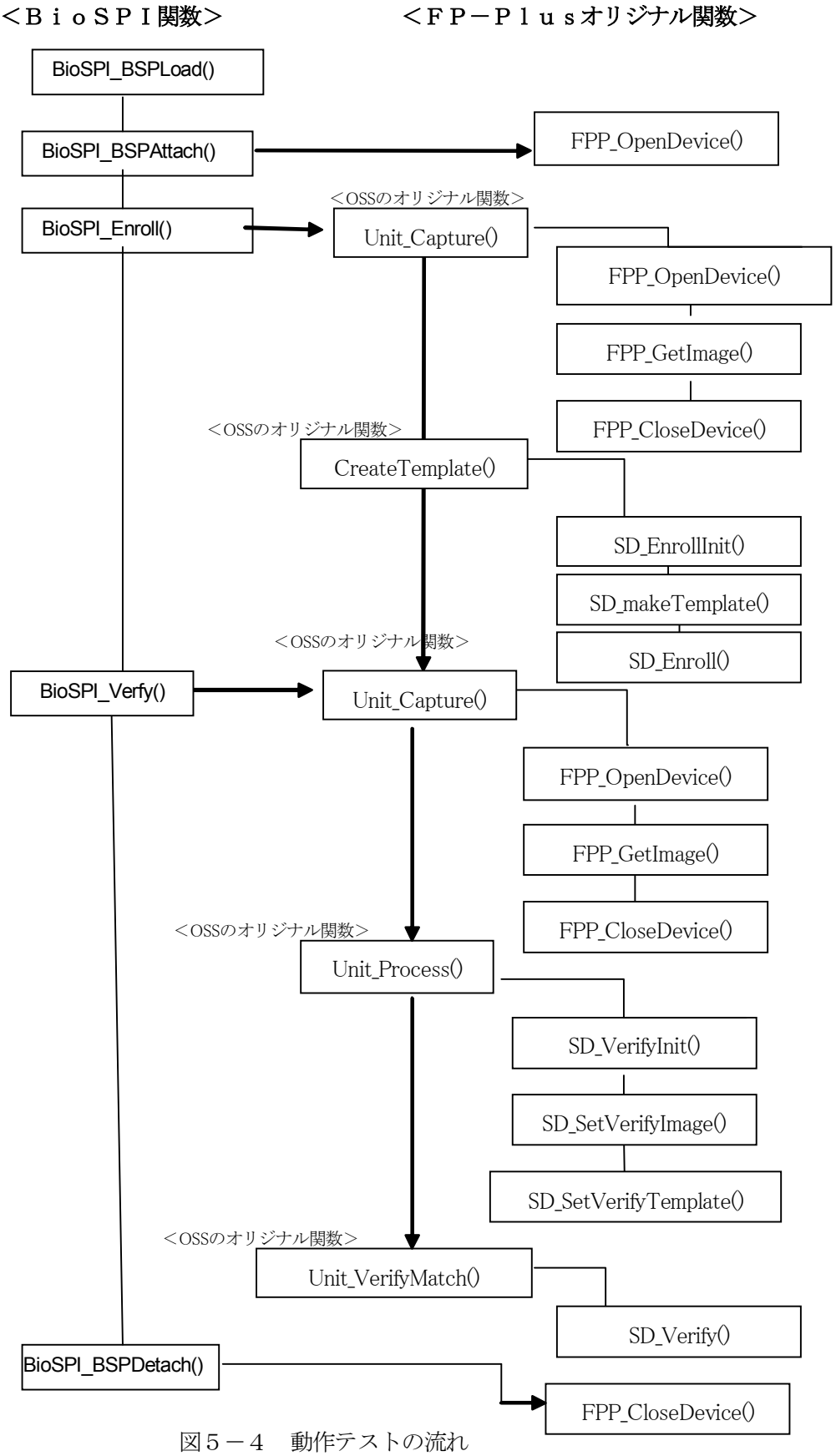


図５－４　動作テストの流れ

（4）ＦＰ－ＰＬＵＳのＳＤＫ関数

今回使用したＦＰ－ＰＬＵＳのＳＤＫ関数仕様を以下にまとめる。

(a) FP-PLUS/MB SDK APIの関数表

デバイスの初期化 / 解放	
FPP_GetDeviceList	デバイスリストの取得
FPP_OpenDevice	デバイスのオープン
FPP_CloseDevice	デバイスのクローズ
FPP_ClearDeviceList	デバイスリストのクリア
デバイス情報 / 状態	
FPP_GetDeviceInfo	デバイス情報の取得
FPP_GetDeviceDetail	デバイス詳細情報の取得
FPP_IsConnected	デバイスが接続されているか否かのチェック
FPP_ResetReconnectedFlag	Reconnected_flgのリセット
デバイス内部領域の利用	
FPP_ReadUserId	ユーザIDの読み込み
FPP_WriteUserId	ユーザIDの書き込み
FPP_ReadTemplate	テンプレートデータの読み込み
FPP_WriteTemplate	テンプレートデータの書き込み
デバイス操作	
FPP_GetImage	指紋画像の取得
FPP_SetTimeout	指紋画像読み取りタイムアウトの設定
指紋認証エンジン	
SD_EnrollInit	指紋登録準備
SD_makeTemplate	指紋テンプレートの生成
SD_Enroll	登録用指紋テンプレートの生成
SD_VerifyInit	指紋照合準備
SD_SetVerifyTemplate	指紋テンプレートを照合用テンプレートバッファにセット
SD_SetVerifyImage	指紋画像を照合用画像バッファにセット
SD_Verify	指紋テンプレートの照合
SDKバージョン情報	
FPP_GetSdkVers	SDKのバージョン情報取得
SD_GetEngineVers	指紋認証エンジンのバージョン情報取得
画像処理補助	
FPP_ShowFpBitmapX	指紋画像の表示
FPP_SaveBitmap	指紋画像をファイルへ保存

（ｂ）各関数の詳細仕様

・FPP_GetDeviceList

【機　能】　デバイスリストの取得

【宣　言】　int　FPP_GetDeviceList (DEVICE_LIST* *pDevList*);

【引　数】　*pDevList*　デバイスリスト格納領域

【戻り値】　成功した場合 FP_PLUS_OK が戻り値となる。
失敗した場合 エラーコードが戻る。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照。

【解　説】　認識されたFP-PLUSデバイスのリストを作成し、*pDevList* へ情報を格納する。
この関数は呼び出される度に、最新のリストを作成する。
一度リストされたデバイスは、物理的に抜かれた後でも未接続のデバイスとしてリストに残る。
リストを最新の情報に更新したい場合には、FPP_ClearDeviceListを呼び出した後に実行すること。
構造体 DEVICE_LIST の詳細は 「注記２．API定数及び構造体」を参照のこと。

【使用例】

```
DEVICE_LIST DevList;
CString strMsg;
// デバイスリストの取得
if( FPP_GetDeviceList(&DevList) != FP_PLUS_OK )
 AfxMessageBox("デバイスリストの取得に失敗しました。");
// デバイスリストの表示
for(i=0; i<DevList.dev_count; i++)
{
 strMsg.Format("デバイスID [%d] = %s", i, DevList.dev_info[i].dev_id);
 AfxMessageBox(strMsg);
}
```

・*FPP_OpenDevice*

【機　能】　FP-PLUSデバイスのオープン

【宣　言】　int　FPP_OpenDevice (LPCSTR *lpcstrDevId*, int* *pDevHandle*);

【引　数】　*lpcstrDevId*　デバイスID
pDevHandle　デバイスハンドルの格納領域

【戻り値】　成功した場合 FP_PLUS_OK が戻り値となる。
失敗した場合 エラーコードが戻る。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　FP-PLUSデバイスをオープンする。
*lpcstrDevId*には、オープンしたいデバイスのデバイスIDをセットする。
*lpcstrDevId*に NULL 又は“”をセットすると最初に認識されたデバイスが

オープン対象となる。
成功した場合は、*pDevHandle*にデバイスハンドルがセットされる。

【使用例】

```
int dev_h;
① デバイスリストを取得し、デバイスIDを指定してオープンする例
DEVICE_LIST dev_list;
// デバイスリストの取得
if( FPP_GetDeviceList(&dev_list) != FP_PLUS_OK )
 AfxMessageBox("デバイスリストの取得に失敗しました。");
// FP-PLUSデバイスのオープン
if( FPP_OpenDevice(dev_list.dev_info[0].dev_id, &dev_h) == FP_PLUS_OK )
        AfxMessageBox("FP-PLUSのオープンに成功しました。");
   ② デバイスID無指定オープンの例
  if( FPP_OpenDevice(NULL, &dev_h) == FP_PLUS_OK )
 AfxMessageBox("FP-PLUSのオープンに成功しました。");
③ デバイスID指定オープンの例
if( FPP_OpenDevice("64df2c0d6df7dccf", &dev_h) == FP_PLUS_OK )
 AfxMessageBox("FP-PLUSのオープンに成功しました。");
```

・FPP_CloseDevice

【機　能】　FP-PLUSデバイスのクローズ

【宣　言】　int　FPP_CloseDevice(int *DevHandle*);

【引　数】　*DevHandle*　デバイスハンドル

【戻り値】　デバイスIDが有効であれば、FP_PLUS_OK が戻される。
デバイスIDが無効であれば、エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　FP-PLUSデバイスをクローズする。

【使用例】

```
// FP-PLUSデバイスのクローズ
if( FPP_ CloseDevice (dev_h) == FP_PLUS_OK )
{
 AfxMessageBox("FP-PLUSのクローズに成功しました。");
}
```

・FPP_ClearDeviceList

【機　能】　デバイスリストのクリア

【宣　言】　int　FPP_ ClearDeviceList ();

【引　数】　なし

【戻り値】　成功した場合FP_PLUS_OK が戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　デバイスリストをクリアする。
オープンされているデバイスは、全てクローズされる。
(*)MAX_DEVICESを超える数のデバイスを入れ替えて利用するような特殊なシステムの場合に利用する。

【使用例】

```
// デバイスリストのクリア
if( FPP_ ClearDeviceList () == FP_PLUS_OK )
 AfxMessageBox("デバイスリストはクリアされました。");
// デバイスリストの取得
if( FPP_GetDeviceList(&DevList) == FP_PLUS_OK )
 AfxMessageBox("デバイスリストの取得に成功しました。");
```

・FPP_GetDeviceInfo

【機　能】　デバイス情報の取得

【宣　言】　int　FPP_GetDeviceinfo(int *DevHandle*, DEVICE_INFO* *pDevInfo*);

【引　数】　*DevHandle*　デバイスハンドル
pDevInfo　デバイス情報の格納領域

【戻り値】　成功した場合FP_PLUS_OK が戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　デバイス情報を取得する。
構造体 DEVICE_INFO の詳細は 「注記２．API定数及び構造体」を参照のこと。

【使用例】

```
DEVICE_INFO dev_info;
// デバイス情報の取得
if( FPP_GetDeviceInfo(dev_h, &dev_info) == FP_PLUS_OK )
{
   // デバイスIDの表示
   AfxMessageBox (dev_info.dev_id);
   // バージョン表示
   AfxMessageBox (dev_info.firm_vers);
      // デバイスが接続されていなければ、接続を要求する。
     if( dev_info.status & MASK_ST1 != ST1_CONNECTED )
           AfxMessageBox("デバイスを接続して下さい。");
```

```
}
```

・*FPP_GetDeviceDetail*

【機　能】　デバイス詳細情報の取得

【宣　言】　int　FPP_GetDeviceDetail(int *DevHandle*, DEVICE_DETAIL* *pDevDetail*);

【引　数】　*DevHandle*　デバイスハンドル

pDevDetail　デバイス詳細情報の格納領域

【戻り値】　成功した場合FP_PLUS_OK が戻される。

失敗した場合 エラーコードが戻される。

エラーコードの詳細は「注記３．API戻り値一覧表」参照下のこと。

【解　説】　デバイスの詳細情報を取得する。

構造体 DEVICE_DETAIL の詳細は 「注記２．API定数及び構造体」を参照のこと。

【使用例】

```
DEVICE_DETAIL dev_detail;
// デバイス詳細情報の取得
if( FPP_GetDeviceDetail (dev_h, &dev_detail) == FP_PLUS_OK )
{
   // 製造年月日の表示
   AfxMessageBox (dev_detail.date_of_manufacture);
   // 製品コードの表示
   AfxMessageBox (dev_ detail.product_code);
      // シリアルNoの表示
    AfxMessageBox (dev_ detail. serial);
   }
```

・*FPP_IsConnected*

【機　能】　デバイスが接続されているか否かのチェックを行う

【宣　言】　int　FPP_ IsConnected (int *DevHandle*);

【引　数】　*DevHandle*　デバイスハンドル

【戻り値】　デバイスが接続されていれば、FP_PLUS_OK が戻される。

無効であれば、それ以外の値が戻される。

エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　デバイスが接続され、認識できているか否かをチェックする。

【使用例】

```
// デバイスが利用可能か否かをチェックする。
if( FPP_ IsConnected(dev_h) != FP_PLUS_OK )
{
 AfxMessageBox("FP-PLUSが接続されていません。");
```

```
}
```

・FPP_ResetReconnectedFlag

【機　能】　Reconnected_flgのリセット

【宣　言】　int　FPP_ ResetReconnectedFlag (int *DevHandle*);

【引　数】　*DevHandle*　デバイスハンドル

【戻り値】　成功した場合FP_PLUS_OK が戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　デバイスが一度抜かれて、再度接続された場合Reconnected_flg に1の値が挿入される。

この関数では、このReconnected_flg を 0 にリセットする。
Reconnected_flg は、GetDeviceInfo()にて取得されるDEVICE_INFO構造体で参照可能である。

【使用例】

```
// デバイス情報の取得
FPP_GetDeviceInfo(dev_h, &dev_info);
// 接続状態の表示
if( GET_ST1(dev_info.status) == ST1_CONNECTED )
{
    if( dev_info.Reconnected_flg) // 再接続？
    {
      AfxMessageBox("再接続されました。");
        // Reconnected_flgを0にリセット
      FPP_ResetReconnectedFlag(dev_h);
    }
    else
        AfxMessageBox ("接続されています。");
}
else
      AfxMessageBox ("デバイスは抜かれました。");
```

・*FPP_ReadUserId*

【機　能】　FP-PLUSデバイス内のユーザIDの読み込み

【宣　言】　int　FPP_ReadUserId(int *DevHandle*, LPTSTR　*lptstrUserId*);

【引　数】　*DevHandle*　デバイスハンドル
lptstrUserId　ユーザIDの格納場所　(DI_USERID_SIZE以上の領域が必要)

【戻り値】　成功した場合 FP_PLUS_OK が戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。
【解　説】　FP-PLUSデバイス内のユーザ領域に登録されたユーザIDを読み込む。
【使用例】

```
// ユーザIDの保管領域
 char   strUserId[DI_USERID_SIZE];
// ユーザIDの読み込み
rc = FPP_ReadUserId( dev_h, strUserId );
 if( rc == FP_PLUS_OK )
      AfxMessageBox("ユーザIDの読み込みに成功しました。");
else
      AfxMessageBox("ユーザIDの読み込みに失敗しました。");
```

・*FPP_WriteUserId*
【機　能】　FP-PLUSデバイス内へのユーザIDの書き込み
【宣　言】　int　FPP_WriteUserId(int *DevHandle*, LPCTSTR　*lpctstrUserId*);
【引　数】　*DevHandle*　　デバイスハンドル
lpctstrUserId　ユーザIDの格納場所
【戻り値】　成功した場合 FP_PLUS_OK が戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。
【解　説】　FP-PLUSデバイス内のユーザ領域にユーザIDを書き込む。
ユーザIDは、16バイト以内となる。
定数DI_USERID_SIZEは、16バイトにnullを含めた17で定義されている。
【使用例】

```
// ユーザID変数宣言
char   strUserId[DI_USERID_SIZE];
// ユーザID設定
strcpy( strUserId, "user id" );
// ユーザIDの書き込み
rc = FPP_WriteUserId(dev_h, strUserId );
 if( rc == FP_PLUS_OK )
      AfxMessageBox("ユーザIDの書き込みに成功しました。");
else
      AfxMessageBox("ユーザIDの書き込みに失敗しました。");
```

・*FPP_ReadTemplate*

【機　能】　FP-PLUSデバイス内のテンプレートデータの読み込みを実行する。

【宣　言】　int　FPP_ReadTemplate(
　　int　　*DevHandle*,
　　int *AreaNo*,
　　BYTE＊　　*TemplateBuff*
　　);

【引　数】　*DevHandle*　　デバイスハンドル
　　AreaNo　　テンプレートエリア番号
　　TemplateBuff　テンプレートの保存場所（TEMPLATE_SIZE以上の領域が必要）

【戻り値】　成功した場合 FP_PLUS_OK が戻される。
　　失敗した場合 エラーコードが戻される。
　　エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　FP-PLUSデバイス内のユーザ領域に登録された指紋テンプレートデータを読み込む。
　　テンプレートエリアは２つ用意されており、１と２が指定できる。

【使用例】

```
BYTE　TemplateBuff[TEMPLATE_SIZE];
int　AreaNo;
  // テンプレートデータの読み込み（エリア１）
  AreaNo = 1;
  rc = FPP_ReadTemplate( dev_h, AreaNo, TemplateBuff );
  if( rc == FP_PLUS_OK )
      AfxMessageBox(“テンプレートの読み込みに成功しました。”);
  else
      AfxMessageBox(“テンプレートの読み込みに失敗しました。”);
```

・*FPP_WriteTemplate*

【機　能】　FP-PLUSデバイス内へのテンプレートデータの書き込み

【宣　言】　int　FPP_WriteTemplate(
　　int *DevHandle*,
　　int　　*AreaNo*,
　　BYTE＊　　*TemplateBuff*
　　);

【引　数】　*DevHandle*　　デバイスハンドル
　　AreaNo　　テンプレートエリア番号
　　TemplateBuff　テンプレートの格納場所（TEMPLATE_SIZE以上の領域が必要）

【戻り値】　成功した場合 FP_PLUS_OK が戻される。
　　失敗した場合 エラーコードが戻される。

エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　FP-PLUSデバイス内のユーザ領域に指紋テンプレートデータを書き込む。
テンプレートエリアは２つ用意されており、１と２が指定できる。

【使用例】

```
BYTE   TemplateBuff[TEMPLATE_SIZE];
int    AreaNo;
  // テンプレートデータの生成処理
  ・・・
  // テンプレートデータの書き込み（エリア１）
  AreaNo = 1;
  rc = FPP_WriteTemplate( dev_h, AreaNo, TemplateBuff );
  if( rc == FP_PLUS_OK )
       AfxMessageBox(“テンプレートの書き込みに成功しました。”);
  else
     AfxMessageBox(“テンプレートの書き込みに失敗しました。”);
```

・*FPP_GetImage*

【機　能】　指紋画像の取得

【宣　言】　int　FPP_GetImage(int *DevHandle*, BYTE **ImgBuff*);

【引　数】　*DevHandle*　　デバイスハンドル
ImgBuff　　指紋画像の格納場所（IMG_SIZE以上の領域が必要）

【戻り値】　成功した場合 FP_PLUS_OK が戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　FP-PLUSデバイスから指紋画像を取得する。
この関数は、FP-PLUSを操作し指紋画像を読み取る。
①FP-PLUS装置上面のLEDが点滅し、指紋読み取り待機中になる。
②センサーのスキャンエリアに指を置くと指紋を読み取る。
③読み取った画像が、引数ImgBuffに渡される。
ImgBuffには、必ずIMG_SIZE※1）以上の領域を用意すること。
取得したImgBuff は、SD_makeTemplate( )への引数として利用できる。
※１）詳細は「注記３．API戻り値一覧表」を参照のこと。
※指紋読み取りについては「FP-PLUS取扱説明書」を参照のこと。

【使用例】

```
BYTE ImgBuff[IMG_SIZE];
switch( FPP_GetImage(dev_h, ImgBuff) ) // 指紋画像の取得
{
```

```
case FP_ PLUS _OK:
    AfxMessageBox ( “指紋画像を取得しました。” );
    break;
case TIME_OUT: // 10秒間待機した後にタイムアウト
    AfxMessageBox ( “指紋画像の取得に失敗しました。（タイムアウト）”);
    break;
default:
     AfxMessageBox ( “指紋画像の取得に失敗しました。”);
    break;
}
```

・*FPP_SetTimeout*

【機　能】　指紋画像読み取りタイムアウトの設定

【宣　言】　int　FPP_SetTimeout(int *DevHandle*, int msec)

【引　数】　*DevHandle*　デバイスハンドル

msec　　タイムアウトまでの時間（ミリ秒）

【戻り値】　成功した場合 FP_PLUS_OK が戻される。

失敗した場合 エラーコードが戻される。

エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　FP- PLUSデバイスから指紋画像を読み取る際のタイムアウトを設定する。

タイムアウトはミリ秒単位で指定する。

初期値　　5000msec

範囲　3000msec～15000msec

【使用例】

```
BYTE ImgBuff[IMG_SIZE];
FPP_SetTimeout(10000);        // 読み取りタイムアウトを10秒間に設定
switch( FPP_GetImage(dev_h, ImgBuff) ) // 指紋画像の取得
{
case FP_ PLUS _OK:
    AfxMessageBox ( “指紋画像を取得しました。” );
    break;
case TIME_OUT: // 10秒間待機した後にタイムアウト
    AfxMessageBox ( “指紋画像の取得に失敗しました。（タイムアウト）”);
    break;
default:
     AfxMessageBox ( “指紋画像の取得に失敗しました。”);
    break;
}
```

・*SD_EnrollInit*

【機　能】　指紋登録準備
【宣　言】　void　SD_EnrollInit();
【引　数】　なし
【戻り値】　なし
【解　説】　指紋登録の準備をする。
指紋の登録処理の先頭で一度呼び出しする。
詳細はApplication Samplesマニュアルを参照し、登録処理手順を確認すること。
【使用例】

```
// 登録の為の初期化処理
SD_EnrollInit();
```

・*SD_makeTemplate*
【機　能】　指紋テンプレートの生成
【宣　言】　int　SD_makeTemplate(BYTE＊*ImageBuff*　);
【引　数】　*ImageBuff*　指紋画像データの格納場所 (IMG_SIZE以上の領域が必要)
【戻り値】　成功した場合 SD_OKが戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。
【解　説】　与えられた指紋画像から指紋テンプレートを生成する。
予めFPP_GetImage( )で取得した画像データを渡し、指紋テンプレートを生成する。
テンプレートのサイズについては「注記３．API戻り値一覧表」を参照のこと。
詳細はApplication Samplesマニュアルを参照し、登録処理手順を確認すること。
【使用例】

```
BYTE ImageBuff[IMG_SIZE];
// 指紋画像の取得
・・・
// 指紋テンプレートの生成
if( SD_makeTemplate(ImageBuff) == SD_OK )
    AfxMessageBox(“指紋テンプレートの生成に成功しました。”);
else
    AfxMessageBox(“指紋テンプレートの生成に失敗しました。”);
```

・*SD_Enroll*
【機　能】　登録用指紋テンプレートの生成
【宣　言】　int　SD_Enroll(*BYTE＊TemplateBuff*) ;
【引　数】　*TemplateBuff*　指紋テンプレートの格納場所(TEMPLATE_SIZE以上の領域が必要)
【戻り値】　成功した場合 SD_OKが戻される。
失敗した場合 エラーコードが戻される。

エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　登録用指紋テンプレートを生成し、取得する。
指紋の特徴データを登録するには、指紋画像を３回採取し、それぞれの画像からSD_makeTemplate( )により、３つの指紋テンプレートを生成しておく必要がある。
テンプレートのサイズについては「注記３．API戻り値一覧表」を参照のこと。
詳細はApplication Samplesマニュアルを参照し、登録処理手順を確認のこと。

【使用例】

```
BYTE TemplateBuff[TEMPLATE_SIZE];

// 指紋テンプレートの生成を３回
・・・
// 登録用指紋テンプレートの生成
if( Sd_Enroll(TemplateBuff) == SD_OK )
    AfxMessageBox(“登録用指紋テンプレートの生成に成功しました。”);
else
    AfxMessageBox(“登録用指紋テンプレートの生成に失敗しました。”);
```

・*SD_VerifyInit*

【機　能】　指紋照合準備
【宣　言】　void　SD_VerifyInit() ;
【引　数】　なし
【戻り値】　なし
【解　説】　指紋照合の準備を実行する。
指紋照合処理の先頭で一度呼び出しする。
詳細はApplication Samplesマニュアルを参照し、照合処理手順を確認下すること。

【使用例】

```
// 指紋照合の為の初期化処理
SD_VerifyInit();
```

・*SD_SetVerifyTemplate*

【機　能】　指紋テンプレートを照合用テンプレートバッファに設定する。
【宣　言】　int　SD_SetVerifyTemplate(BYTE＊*TemplateBuff*) ;
【引　数】　*TemplateBuff*　指紋テンプレートの格納場所(TEMPLATE_SIZE以上の領域が必要)
【戻り値】　成功した場合 SD_OKが戻される。

失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　指紋テンプレートを照合用テンプレートバッファにセットする。
テンプレートとテンプレートを照合する際に利用する。
詳細はApplication Samplesマニュアルを参照し、照合処理手順を確認すること。

【使用例】

```
BYTE TemplateBuff[TEMPLATE_SIZE];
  // テンプレートを照合元にセットする。
  if( SD_SetVerifyTemplate(TemplateBuff) == SD_OK )
      AfxMessageBox(“照合元テンプレートのセットに成功しました。”);
  else
      AfxMessageBox(“照合元テンプレートのセットに失敗しました。”);
```

・*SD_SetVerifyImage*

【機　能】　指紋画像を照合用画像バッファにセット

【宣　言】　int　SD_SetVerifyImage(BYTE＊*ImageBuff*　);

【引　数】　*ImageBuff*　指紋画像の格納場所（IMG_SIZE以上の領域が必要）

【戻り値】　成功した場合 SD_OKが戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　指紋画像を照合用画像バッファにセットする。
指紋画像とテンプレートを照合する際に利用する。
詳細はApplication Samplesマニュアルを参照し、照合処理手順を確認すること。

【使用例】

```
BYTE ImageBuff[IMG_SIZE];
  // 指紋画像を照合元にセットする。
  if( SD_SetVerifyImage(ImageBuff) == SD_OK )
      AfxMessageBox(“照合元指紋画像のセットに成功しました。”);
  else
      AfxMessageBox(“照合元指紋画像のセットに失敗しました。”);
```

・*SD_Verify*

【機　能】　指紋テンプレートの照合

【宣　言】　int　SD_Verify(BYTE＊*TemplateBuff*　)

【引　数】　*TemplateBuff*　　照合先テンプレートの格納場所
（TEMPLATE_SIZE以上の領域が必要）

【戻り値】　成功した場合 SD_OKが戻される。

失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　予めセットした照合元テンプレートと照合先テンプレートを照合する。
詳細はApplication Samplesマニュアルを参照し、照合処理手順を確認すること。

【使用例】

```
BYTE TemplateBuff[TEMPLATE_SIZE];
  // 指紋を照合する。
  if( SD_Verify(TemplateBuff) == SD_OK )
      AfxMessageBox("同一指紋として照合されました。");
  else
      AfxMessageBox("照合されません。"); // 別の指紋
```

・*FPP_GetSdkVers*

【機　能】　SDKのバージョン情報取得

【宣　言】　int　FPP_GetSdkVers(
LPTSTR *lpszProduct*,
LPTSTR *lpszVers*
)

【引　数】　*lpszProduct*　　プロダクトIDの格納場所 (SDK_CODE_SIZE以上の領域が必要)
lpszVers　　バージョンの格納場所 (SDK_VERS_SIZE以上の領域が必要)

【戻り値】　成功した場合 FP_PLUS_OKが戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　SDKのバージョン情報を取得する。
バージョン情報の例：
プロダクトID：　**PSM-VXX-SDD-01/2007**
バージョン：　**1.6.0.0**

【使用例】

```
char   strProduct[SDK_CODE_SIZE];
char   strVers[SDK_VERS_SIZE];
// SDKデバイスコントローラのバージョン
if( FPP_GetSdkVers(strProduct, strVers) == FP_PLUS_OK)
{
    AfxMessageBox( strProduct );
    AfxMessageBox( strVers );
}
```

・*SD_GetEngineVers*

【機　能】　指紋認証エンジンのバージョン情報取得

【宣　言】　int　SD_GetEngineVers(
LPTSTR *lpszProduct*,
LPTSTR *lpszVers*
)

【引　数】　*lpszProduct*　　プロダクトIDの格納場所 (SDK_CODE_SIZE以上の領域が必要)
lpszVers　　バージョンの格納場所 (SDK_VERS_SIZE以上の領域が必要)

【戻り値】　成功した場合 SD_OKが戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　指紋認証エンジンのバージョン情報を取得する。
バージョン情報の例：
認証エンジン：　PSM-VXX-SED-01/2007
バージョン：　1.0.2.10209

【使用例】

```
 char   strProduct[SDK_CODE_SIZE];
char   strVers[SDK_VERS_SIZE];
// 指紋認証エンジンのバージョン
if( SD_GetEngineVers(strProduct, strVers) == SD_OK)
{
    AfxMessageBox( strProduct );
    AfxMessageBox( strVers );
}
```

・*FPP_ShowFpBitmapX*

【機　能】　指紋画像の表示

【宣　言】　int　FPP_ShowFpBitmapX(
HDC *hDC*,
BYTE＊ *ImgBuff*,
int *width*,
int *height*
);

【引　数】　*hDC*　　デバイスコンテキストへのハンドル
ImgBuff　指紋画像の格納場所
width　　指紋画像の表示幅
height　　指紋画像の表示高さ

【戻り値】　成功した場合 FP_PLUS_OKが戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　デバイスコンテキストを指定し、指紋画像を表示する。

FPP_GetImage( )により取得した画像データを、任意のサイズで表示する。

【使用例】

```
// 指紋画像を200×240で表示
HDC hDC = m_FpImage.GetDC()->m_hDC; // 表示先デバイスコンテキスト
if( FPP_ShowFpBitmapX(hDC, ImageBuff, 200, 240) == FP_PLUS_OK )
    // 指紋画像が表示された。
eles
{
    // 指紋画像の表示に失敗した。
    AfxMessageBox("指紋画像の表示に失敗しました。");
}
```

・*FPP_SaveBitmap*

【機　能】　指紋画像をファイルへ保存

【宣　言】　int　FPP_SaveBitmap(
LPCSTR *lpszFileName* ,
BYTE **ImgBuff*
)

【引　数】　*lpszFileName*　　保存するビットマップファイル名
ImgBuff　　指紋画像格納場所

【戻り値】　成功した場合 FP_PLUS_OKが戻される。
失敗した場合 エラーコードが戻される。
エラーコードの詳細は「注記３．API戻り値一覧表」を参照のこと。

【解　説】　指紋画像をビットマップファイルに保存する。
FPP_GetImage( )により取得した画像データをBMPファイルへ保存する。
例：ドライブCのルートにa.bmpというファイル名で保存したい場合には
引数FileNameに"C:¥¥a.bmp"と指定すること。

【使用例】

```
// 指紋画像をビットマップファイルに保存
if( SaveBitmap(strFileName, ImageBuff) == FP_PLUS_OK )
    AfxMessageBox( "画像をビットマップファイルにセーブしました。" );
```

注記１．API宣言一覧

```
// デバイスの初期化 / 解放
extern int      FPP_GetDeviceList(DEVICE_LIST *dev_list);
extern int      FPP_OpenDevice(LPCSTR lpcstrDID, int *DevHandle);
extern int      FPP_CloseDevice(int DevHandle);
extern int      FPP_ClearDeviceList(void);
// デバイス情報 / 状態
```

```
extern int          FPP_GetDeviceInfo(int DevHandle, DEVICE_INFO *dev_info);
extern int          FPP_GetDeviceDetail(int DevHandle, DEVICE_DETAIL *dev_detail);
extern int          FPP_IsConnected(int DevHandle);
extern int          FPP_ResetReconnectedFlag(int DevHandle);
// デバイス内部領域の利用
extern int          FPP_ReadUserId(int DevHandle, LPTSTR lpstrUserId);
extern int          FPP_WriteUserId(int DevHandle, LPCTSTR lpstrUserId);
extern int          FPP_ReadTemplate(int DevHandle, int , BYTE *) ;
extern int          FPP_WriteTemplate(int DevHandle, int , BYTE *) ;
// デバイス操作
extern int          FPP_GetImage(int DevHandle, BYTE *) ;
extern int          FPP_SetTimeout(int DevHandle, int) ;
// 指紋認証エンジン
extern void         SD_EnrollInit() ;
extern int          SD_makeTemplate(BYTE *) ;
extern int          SD_Enroll(BYTE *) ;
extern void         SD_VerifyInit() ;
extern int          SD_SetVerifyTemplate(BYTE *) ;
extern int          SD_SetVerifyImage(BYTE *) ;
extern int          SD_Verify(BYTE *) ;
// SDKバージョン情報
extern int          FPP_GetSdkVers(LPTSTR lpszProduct,   LPTSTR lpszVers);
extern int          SD_GetEngineVers(LPTSTR lpszProduct,   LPTSTR lpszVers);

// 画像処理補助
extern int          FPP_ShowFpBitmapX(HDC hDC, BYTE *Image, int width, int height);
extern int          FPP_SaveBitmap(LPCTSTR lpszFileName , BYTE *Image);
```

注記２．API定数及び構造体

[データサイズに関する定数]

```
#define IMG_SIZE                48000    // 指紋画像の格納サイズ
#define TEMPLATE_SIZE           200      // 指紋テンプレートの格納サイズ
#define DI_DATE_SIZE            11       // デバイス・製造年月日の格納サイズ (“yyyy/mm/dd”)
#define DI_CODE_SIZE            9        // デバイス・製品コードの格納サイズ (末尾はnull)
#define DI_SERIAL_SIZE          9        // デバイス・シリアルNoの格納サイズ (末尾はnull)
#define DI_USERID_SIZE          17       // デバイス・ユーザIDの格納サイズ (末尾はnull)
#define DI_ID_SIZE          17       // デバイス・IDの格納サイズ (末尾はnull)
#define DI_FWVERS_SIZE          19       // デバイス・Firmware versionの格納サイズ (末尾はnull)
```

```
#define SDK_CODE_SIZE           21      // SDK・プロダクトコードの格納サイズ（末尾はnull）
#define SDK_VERS_SIZE           21      // SDK・プロダクトバージョンの格納サイズ（末尾はnull）
#define MAX_DEVICES             64      // 認識可能な最大デバイス数
[デバイスの接続状態に関する定数（ステータス１）]
#define ST1_DISCONNECTED        0x00    // 未接続 （もしくは、他のプロセスで利用している）
#define ST1_CONNECTED           0x10    // 接続
[デバイスの利用状況に関する定数（ステータス２）]
#define ST2_AVAILABLE           0x01    // 空き（利用可）
#define ST2_USED                0x02    // 利用中
[デバイスステータスのマスク]
#define MASK_ST1                0xf0    // ステータス１のマスク
#define MASK_ST2                0x0f    // ステータス２のマスク

[デバイスステータス取得補助用のインライン関数]
#define GET_ST1(status) (status & MASK_ST1)    // ステータス１の取得
#define GET_ST2(status) (status & MASK_ST2)    // ステータス２の取得

// デバイス情報
typedef struct {
        int     dev_handle;                     // デバイスハンドル
        char    dev_id[DI_ID_SIZE];             // デバイスID
        char    firm_vers[DI_FWVERS_SIZE];      // ファームウェア・バージョン
        int     status;                         // デバイス利用状況ステータス(ST1_??? + ST2_???)
        int     timeout_msec;                   // 指紋画像取得タイムアウト（msec）
        int     reconnected_flg;                // デバイスが再接続されると1 が入る
}DEVICE_INFO;

// デバイスリスト
typedef struct {
        int             dev_count;              // 利用可能なデバイスの数
        DEVICE_INFO     dev_info[MAX_DEVICES];  // デバイス情報のリスト
}DEVICE_LIST;

// デバイス詳細情報
typedef struct {
        char    dev_id[DI_ID_SIZE];             // デバイスID（NULLで終わる文字列）
        char    firm_vers[DI_FWVERS_SIZE];      // デバイスのバージョン文字列（型番とバージョン）
        double  firm_vers_num;                  // デバイスのバージョン番号
```

```
    char    date_of_manufacture[DI_DATE_SIZE];    // デバイスの製造年月日
    char    product_code[DI_CODE_SIZE];// デバイスの製品コード
    char    serial[DI_SERIAL_SIZE];               // デバイスのシリアルNo
}DEVICE_DETAIL;
```

注記３．API戻り値一覧表

```
#define FP_PLUS_OK                  0       // 成功
#define OPEN_ERR                    1       // FP-PLUSﾓｼﾞｭｰﾙのｵｰﾌﾟﾝ失敗
#define RESET_ERR           2       // FP-PLUSﾓｼﾞｭｰﾙのﾘｾｯﾄ失敗
#define TIMEOUT_SET_ERR             3       // FP-PLUSﾓｼﾞｭｰﾙのﾀｲﾑｱｳﾄの設定失敗
#define PURGE_ERR                   4       // FP-PLUSﾓｼﾞｭｰﾙのﾊﾞｯﾌｧのﾊﾟｰｼﾞ(ｸﾘｱ)失敗
#define CLOSE_ERR                   5       // FP-PLUSﾓｼﾞｭｰﾙのｸﾛｰｽﾞ失敗
#define GETIMAGE_ERR                6       // 指紋画像読み取り失敗
#define GETIMAGE_CANCEL 7       // 指紋画像読み取りｷｬﾝｾﾙ
#define IMAGE_RECOGNIZE_ERR         8       // 指紋画像検出失敗
#define TIMEOUT_VALUE_ERR           9       // ﾀｲﾑｱｳﾄ値の範囲ｴﾗｰ(3000～15000)
#define DATANO_ERR                  10      // 指定したﾃﾞｰﾀ番号に誤りがある
#define WRITE_ERR           11      // ﾃﾞｰﾀの書き込み失敗
#define READ_ERR                    12      // ﾃﾞｰﾀの読み込み失敗
#define PROTECT_ERR                 13      // ﾗｲﾄﾌﾟﾛﾃｸﾄがｾｯﾄされている領域に書き込もうとした
#define DID_ERR                     14      // FP-PLUSのﾃﾞﾊﾞｲｽIDが未登録
#define MEMORY_ALLOC_ERR            15      // ﾒﾓﾘの確保に失敗
#define FILE_OPEN_ERR               16      // ﾌｧｲﾙのｵｰﾌﾟﾝ失敗
#define FILE_WRITE_ERR              17      // ﾌｧｲﾙの書き込み失敗
#define TIME_OUT                    18      // ﾀｲﾑｱｳﾄ
#define INVALID_HANDLE              40      // デバイスハンドルが不正
#define NOT_RECOGNIZED              41      // デバイスが認識されない
#define ALREADY_OPENED              42      // 既にオープンされているデバイスをオープンしようとした
#define DISCONNECTED                43      // デバイスが抜かれた

#define SD_OK                       0       // 成功。
#define SD_DIFF                     1       // 照合している二つのﾃﾝﾌﾟﾚｰﾄが違う指から採られた。
#define SD_SEQERR           2       // 関数を呼び出す順序が違うための失敗。
#define SD_BADFP                    3       // 指紋に問題があるための失敗。
#define SD_BADTPL                   4       // ﾃﾝﾌﾟﾚｰﾄに問題があるための失敗。
#define SD_BUG                      5       // 内部バグによる失敗。問い合わせが必要。
```

５．４　テスト結果

ＯＳＳ社が用意したフレームワーク用テストプログラムの動作は、ＯＳＳが用意したサンプルＢＳＰ（パスワード認証）を用いた動作テスト、およびセキュアデザイン株式会社の指紋収集装置ＦＰ－ＰＬＵＳを利用した簡易型ＢＳＰを製作して動作を確認した。結果としては、スムーズに動作し問題は確認されなかった。このテストを基にして、フレームワークのＢｉｏＡＰＩ２．０への適合性試験用プログラムの製作を開始することが可能な状態になったと判断する。

５．５　今後の予定

今回のテストでは、実際のバイオメトリックセンサーとして指紋収集装置を利用したが、ハンドル処理、イベント処理等の機能については、テストが未了である。このため、まず、今回、実施出来なかった機能の確認テストをまず実施する。また、基本動作の確認が出来たため、フレームワークの適合試験用プログラムの構成要素となる適合試験のアサーション（５節参照）をプログラム化し、動作の確認を行う予定である。

６　ＢｉｏＡＰＩをはじめとするＳＣ３７の適合性評価の実運用方法の調査

６．１　調査研究の目的

６．１．１　目的

市場のグローバル化が進み，あらゆる製品，サービス，プロセスが国境を越え，世界中に流通する状況において，国際標準に準じることが，世界経済および自国経済の円滑な発展に重要である。

標準，規格，規定が制定された場合，製品，サービス，プロセスが制定した規格などを満たしているか否かの確認が必要となる。この国際標準に準拠した確認を適合性評価(Conformity assessment)という。

バイオメトリック技術は，政府関連機関，地方自治体，金融機関，企業など社会基幹系のユーザに対して導入され，対象システムに対応したバイオメトリック認証製品が開発されている。今後，バイオメトリック認証製品は国際規格に準拠することが，国内基幹システムおよび海外展開する場合，必要になると考える。

従来，「認定」あるいは「認証」という用語は，製品やサービスに対して権威ある機関が，いわゆる「お墨付き」を与える場合に用いていた。現在，国際的な標準化機関であるISO内に設置されているCASCO(適合性評価委員会)*において，その基本的な枠組みの取り決めが検討されている。

本調査では，以下の観点で，日本におけるISO/IEC JTC1１/SC37に関する適合性評価の体制を調査検討する。

⑴ 適合性評価制度に関し，公開情報(ISOドキュメント，日本規格協会，米国NIST，韓国K-NBTC，各カンファレンス資料ほか)より分析整理する。

⑵ 適合性評価制度の実体を，国際標準のフレームワークおよび海外の先行事例などを調査することにより明らかにする。

⑶ 日本における体制整備に関し，日本の産業力強化の観点から戦略的なシナリオを策定する。

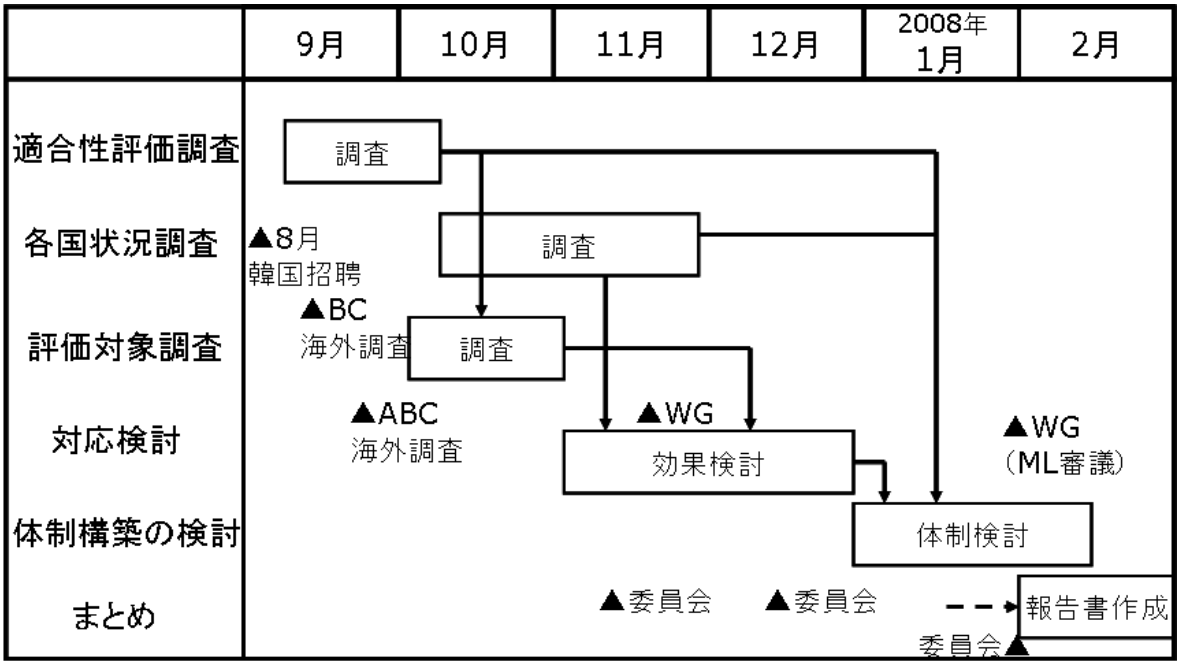
＊適合性調査委員会ISO/CASCO（Committee on Conformity Assessment）は，国際標準化機構ISO（International Organization　for　standardization）の政策開発委員会である。

６．１．２　スケジュール

表6-1に示すスケジュールで対応した。日本自動認識システム協会バイオメトリックセキュリティコンソーシアム内にワーキンググループを設置して，産業界の意見収集を行った。

海外調査は，韓国KISA N-NBTCの責任者を東京に招聘し，韓国における適合性評価体制について講演およびヒアリングした。また米国ボルチモアで開催のBiometric Consortium Conference 2007，およびシンガポールで開催したAsia Biometric Consortium 2007に参加し，情報の収集を行った。

表6-1　調査研究のスケジュール



６．２　適合性評価制度の概要

６．２．１　適合性評価制度について

（1）　定義

適合性評価は，適合性調査委員会ISO/CASCO（Committee on Conformity Assessment）により2004年に定められたISO/IEC17000規格『Conformity assessment − Vocabulary and general principles（適合性評価－用語及び一般原則）』によれば，「製品，プロセス，システム，要員または機関に関する規定要求事項が満たされていることの実証」と定義される（付表参照）［1］。

日本工業標準調査会JISCのサイト上（www.jisc.go.jp）では，「一般に，標準・規格・規定といったものが必要とされる場合，さらに，その標準・規格・規定を，製品・サービス・プロセスといったものが満たしているかどうかを確認する行為」と記されている[2]。

（2）　適合性評価のフレームワーク

製品，プロセス，システム，要員または機関に関する規定要求事項が満たされていることを実証するために，具体的には認定（Accreditation）を受けた第三者認証機関(Certification Body)によって認証（Certification）が行われる。

図6-1に示すように，適合性評価とは，ある製品，プロセス，システムなどについてこれまでの認証主体であった供給者や使用者に変わって，認定を受けた第三者が認証を行う制度である。

ISO/IEC17000規格における定義を以下に示す［1］。

１）認定（Accreditation）：適合性評価機関に関し，適合性評価実務を行う能力を公式に実証したことを伝える第三者証明

２）認証（Certification）：製品，プロセス，システム，または要員に関して特定の要求事項に適合していることを第三者が文書で保証する手続き　ただし，製品品質を保証するための規格であるISO9000シリーズのようなシステム規格への適合性保証には，認証の代わりに審査登録（Registration）という用語を用いることもある。

３）第三者適合性評価活動（third-party conformity assessment activity）：対象を提供する人，または組織および，その対象について使用者側の利害を持つ人，または組織の双方から独立した人，または機関によって実施される適合性評価

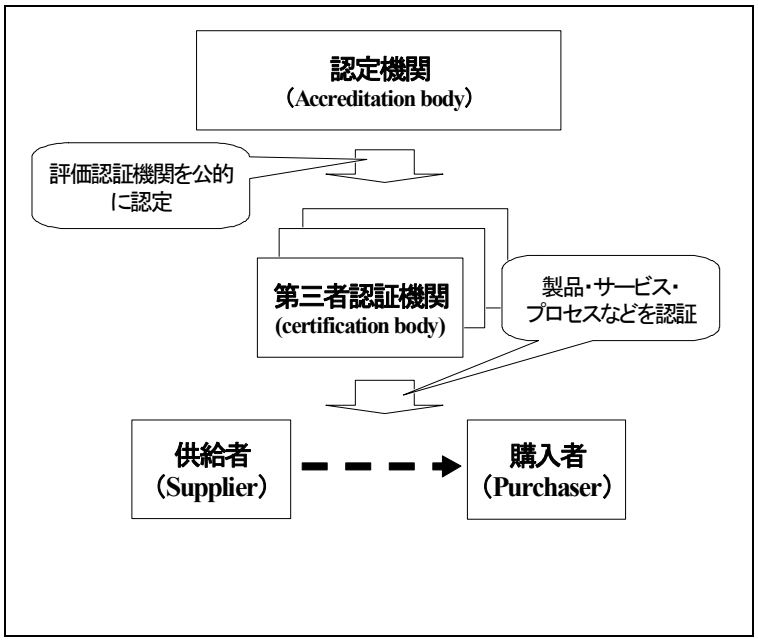


図6-1　適合性評価のフレームワーク

６．２．２　適合性評価統一の国際動向

（1）　従来の適合性評価と問題点

経済活動のグローバル化に伴って、各国の中央政府あるいは地方政府が，国民の生命，財産の保護を目的として，法律により特定の製品やサービスの特性を規定する強制規格（基準認証制度）および，供給者と購入者の間で製品やサービスに対して定められる任意規格分野での購入者側による第二者評価活動が，不必要な貿易障害であり過重な資源，コスト負担の要因であるとして関税および貿易に関する一般協定GATT（General Agreement on Tariffs and Trade）において問題とされていた。

表6-2に強制規格と任意規格についてまとめる。それぞれの規格は適合性評価を行う上で，以下のような問題を抱えている。

- 強制規格は，その国の所轄官庁が必要性に応じて定めることが多く，国際的な整合化に配慮がなされていない。また，規格作成の意図やその過程が取引相手国にとって不透明であること。
- 任意規格は，購入の仕様が購入者の必要に応じて決まるため，要求事項や評価の手続きが購入者ごとに異なり，整合化されていないために供給先ごとに評価を受ける必要があること。

上記のような問題の結果，同一の製品，サービスなどを複数の国へ輸出する供給者は，輸出先ごと，あるいは購入者ごとに基準認証を受けなければならないため適合性評価の透明性・公正性に大きな問題があった[3]。

表6-2　強制規格と任意規格

規 格	内 容	例
任意規格	法律に基づく強権によらず,契約と市場原理に基づいて供給者と購入者の間で製品やサービスに対して定められる規格	JIS（Japanese Industrial Standards）規格など。ただし法的に遵守を義務づけた場合は強制規格となる。
強制規格	各国の中央政府あるいは地方政府が,国民の生命・財産を危険や危害から守るために,法律に基づく強権のもとで特定の製品やサービスの特性を規定し，その遵守を義務づけている規格	電気用品安全法によって規格の遵守が定められているPSE（Product Safety Electrical Appliance & Materials の略）マークなど。

適合性評価活動の主体から考慮すると，表6-3に示すような3つのフレームワークがある。

1）第一者適合性評価（first-party conformity assessment activity）

対象を提供する人，または組織によって実施される適合性評価であり，供給者適合宣言により，決定される。日本においては，ISO14000シリーズなどの認証コストを節約するための手段として，一部の中小企業や自治体が利用している例がみられる。

2）第二者適合性評価　（second-party conformity assessment activity）

対象について使用者側の利害を持つ人，または組織によって実施される適合性評価であり，ある分野で権力をもつ購入者により標準が決まる。例えば，米国のNASA規格や日本の過去の電電公社仕様などが相当する。図6-2に第二者適合性評価活動の例を示す。

3）第三者適合性評価　（third-party conformity assessment activity）

対象を提供する人，または組織およびその対象について使用者側の利害を持つ人，または組織の双方から独立した人，または機関によって実施される適合性評価である。

認定を受けた第三者による認証である。例えば，（財）日本適合性認定協会（JAB：The Japan Accrediation Board for Conformity Assessment）から認証機関として認定された（株）電子商取引安全技術研究所（ECSEC；Electronic Commerce Security Technology Laboratory Inc.）や（財）日本情報処理開発協会（JIPDEC；Japan Information Processing Development Corporation:）などがISO/IEC 15408あるいはISO/IEC　27001の認証を行う仕組みが相当する。

表6-3　適合性評価活動の主体

適合性評価の種類	ISO/IEC17000規格による定義	評価主体	具体例
第一者適合性評価（first-party conformity assessment activity）	対象を提供する人，または組織によって実施される適合性評価	供給者	なし
第二者適合性評価（second-party conformity assessment activity）	対象について使用者側の利害を持つ人，または組織によって実施される適合性評価	購入者	MIL, NASA Technical Standardsなど
第三者適合性評価（third-party conformity assessment activity）	対象を提供する人，または組織およびその対象について使用者側の利害を持つ人，または組織の双方から独立した人，または機関によって実施される適合性評価	認定を受けた第三者	JABやNITEから認定を受けた機関

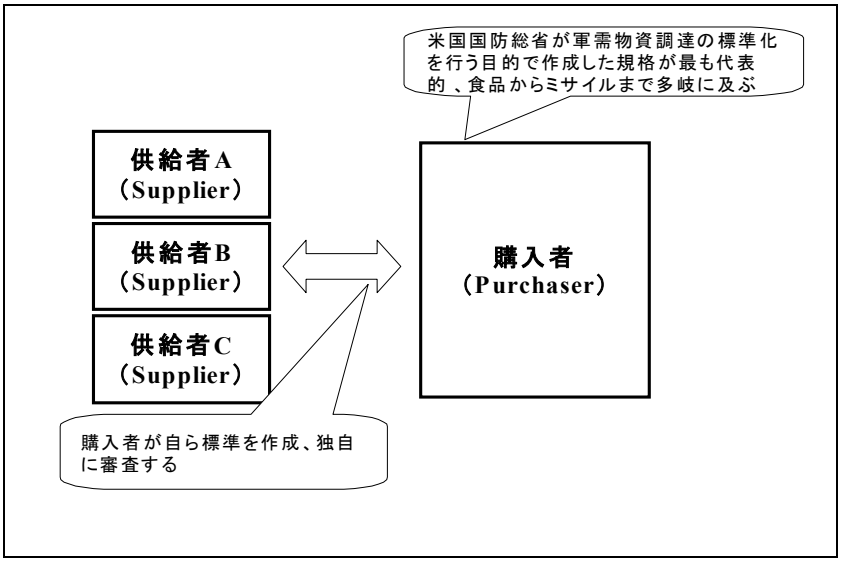


図6-2　第二者適合性評価活動の例

（2）　適合性評価基盤となっている欧州規格

旧欧州共同体EC(European Community)理事会は，1985年に「技術的調和と基準に関するニューアプローチ」(New Approach to technical harmonization and standards)を採択した。

このニューアプローチにより，１つの加盟国の基準認証に適合した製品には安全マークCEマーキング*を貼付し，EC域内全てに出荷流通を認めるなど，各国毎に異なる基準・認証制度を調和させ，EC域内取引の技術的障害を除去するために製品，分野ごとに必須要求事項を整備した。

さらに1989年には「認証とテストに関するグローバルアプローチ」(a global approach to conformity assessment)を議決し，適合性評価機関を認定する機関（試験所，製品認証機関等）をNB（Notified Body：通知機関）として指定し，その適合性評価の結果は域内各国で有効なものすることや，EC域外諸国との適合性評価の相互承認を推進することを盛り込み，EC域内に流通する製品の安全，健康面での規制を制度のもとに統一した［4］。

このニューアプローチ，グローバルアプローチの実現のための規格作成を担当したのが欧州標準化委員会　CEN(Comite Europeen de Normalisation.),および欧州電気標準化委員会 CENELEC（Comite Europeen de Normalisation Electrotechnique）である。CEN,CENELECが作成した規格は欧州規格 EN（European Norm）として，ヨーロッパ連合EU（European Union）地域における統一規格となっている。特に適合性評価手続きに関する規格は，その後ISOの規格に採用され，国際標準の基礎となっている［3］。

＊補足：CEとは，欧州共同体（1967年に6ヶ国で発足した欧州共同体(EC:European Comunities)。その後拡大し，1994年にEU（欧州連合）に発展）のラテン系言語表記に由来。仏語の場合はCommunauté Européenneの頭文字に相当。

（3）　WTOの設立とTBT協定のインパクト

国際貿易の舞台においては，1980年に発行した「関税および貿易に関する一般協定GATTスタンダードコード」で多国間貿易における適合性評価の相違などの非関税障壁を取り除く努力を求めていたが，GATTは一般協定であり国際条約のような法的拘束力がないため実効性を欠いていた。

1995年にGATTを発展させる形で設立された世界貿易機関 WTO(World Trade Organization)は協定ではあるが、基本的な部分については加盟国に対して法的拘束力を課している。

WTOは，貿易の技術的障害に関する協定 WTO/TBT協定（Agreement on Technical Barriers to Trade）において,貿易の技術的障害として規格や適合性評価を上げ，オープンで公平性,公開性,透明性が担保される事を求めている。

具体的には

- 強制規格または任意規格について，国際規格がすでにあるか,または準備中である場合，その規格を基礎とすること
- 適合性評価手続きについて，国際標準化機関によって発表された関連する指針，もしくは規格に基づいて制定すること

などを定めている[3]。

この協定によってWTO加盟国には，ISOなどの国際規格やこれを基準とした適合性評価制度の使用責務が法的根拠を持って発生することとなった。ただし，国民の安全,健康を守ること，環境を守ること，国の安全保障に関わることなどはWTO /TBT協定においても国際ルールの強制はしないことになっている。

以上述べた欧州および世界の適合性評価統一化の流れについて図6-3にまとめる。

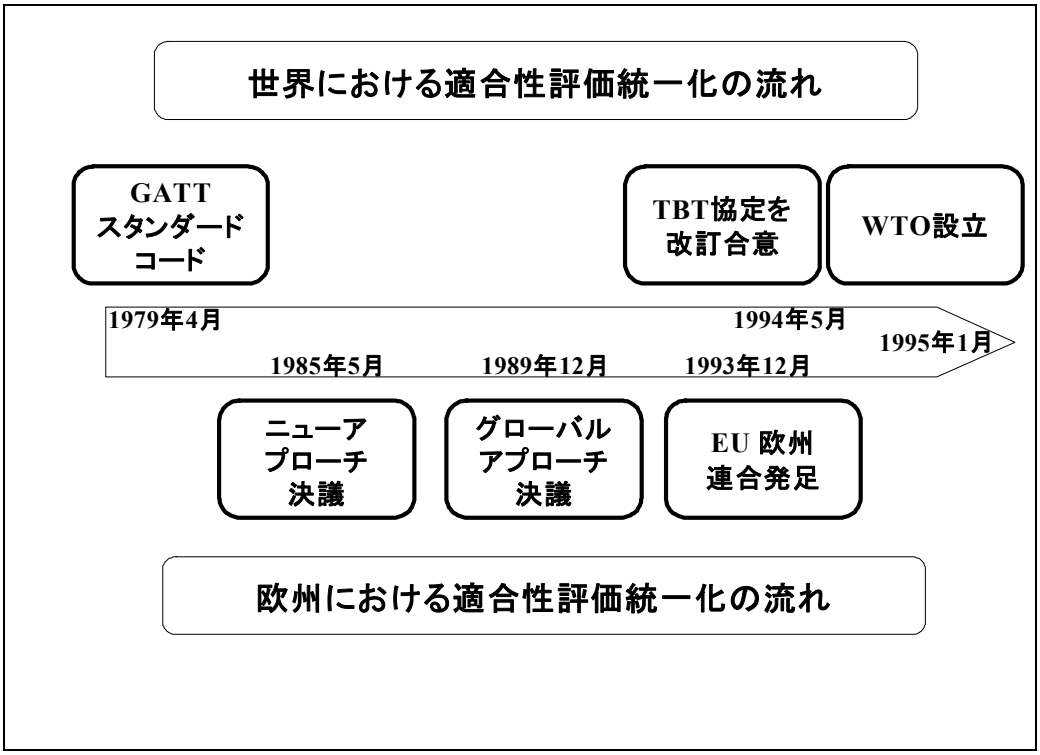


図6-3　適合性評価統一化に関する経緯

（3）　国際相互承認

適合性評価手続の相互承認MR (Mutual Recognition)とは，相互承認の参加機関が，他の参加機関の適合性評価結果を，自ら実施したものと同等であるとして相互に承認するものである[5]。

相互承認は，その参加機関により，以下の３種類に分類する。

1）　政府間相互承認
相手国において行われた自国の強制法規の技術基準への適合性評価の結果が，自国において行われたものと同等であると政府が相互に認め合い，その結果を受け入れること。

2）　認定機関間相互承認
政府間相互承認のように相互の適合性評価の結果の受入れを保証するものではなく，認定機関相互の認定行為の技術的同等性を認定機関同士が確認することを意味する。政府が適合性評価結果を受入れるものではない点で根本的に異なる。

3）　適合性評価機関間相互承認
試験，認証の技術的同等性を適合性評価機関が相互に確認し合い，かつ適合性評価結果を受け入れることを意味する。ただし認定機関間相互承認と同様に政府がその結果を受け入れるものではない。

政府間相互承認の取決めを，相互承認協定MRA(Mutual Recognition Agreement)と呼ぶ。相互承認協定が締結されることにより，従来輸入国内で適合性評価を受ける必要があったものが，当該手

続を輸出国内で実施できるようになることから，適合性評価に伴う手続などの簡素化，コストの削減が見込まれ，貿易の円滑化につながるとされている。WTO/TBT協定の第6.3条においても，WTO加盟国は相互承認の合意のための交渉を行うことが奨励されている[5]。

非政府間レベルであることを明確にする際には，「相互承認取決め」という。

多国間承認取決めMLA(Multilateral Recognition Arrangement)は，複数の機関による非政府間レベルの相互承認取決めである。非政府間レベルのMRAと同義でIAFやPACにおいて使用される[6]。

相互承認協定の具体例として日本では，電気通信機器，電気製品，化学品，医薬品の４分野において日・欧州共同体相互承認協定，電気製品，通信端末機器，無線機器の２分野において日・シンガポール間相互認証協定，日米相互承認協定が結ばれている[7]。

非政府間相互承認の例として大規模なものでは，マネジメントシステム審査登録機関や製品認証機関等を認定するための国際機関である国際認定機関フォーラムIAF（International Accreditation Forum）や，各国における試験所,校正機関,検査機関の認定を行っている試験所認定機関相互の協力機関である，国際試験所認定協力機構ILAC(International　Laboratory Accreditation Cooperation)における協定がある[8]。

６．２．３　国際標準と適合性評価委員会CASCO

（１）　国際標準に関する組織CASCO

WTO/TBT協定は，適合性評価の手続きを，国際標準化機関が発表した指針，または勧告に基づいて作成することを要求している。この「国際標準化機関が発表した指針または勧告」を作成している代表的な機関が国際標準化機構ISO（International Organization for Standardization）である。図6-4に国際標準化機構ISOの体制図を示す[9]。

適合性に関するISO/IEC規格・ガイドは適合性評価委員会 ISO/CASCO（Committee on Conformity Assessment）が審議し制定している。

CASCOは，ISOの組織内における３つの政策開発委員会のうちのひとつで，適合性評価に関する規格や指針の策定を行っている。国際電気標準会議 IEC（International Electrotechnical Commission），国際認定機関フォーラム IAF（International Accreditation Forum），国際試験所認定協力機構 ILAC（International Laboratory Accreditation Cooperation），など多数の適合性評価に関する国際機関の代表者が参加している。

CASCOの具体的な委任事項は以下の通りである[9]。

- 適切な規格，または他の技術仕様に対する製品，プロセス，サービス，マネジメントシステムの適合性評価方法の検討
- 製品，プロセス，サービスの試験，検査，認証の実施，ならびにマネジメントシステム，試験所，検査機関，認証機関，認定機関の評価，及びそれらの運営意図受け入れに関する国際ガイド及び国際規格の作成
- 国家及び地域の適合性評価システムの相互的な認証と受け入れ，ならびに試験，検査，認証，評価，及びそれらに関連する目的のための国際規格の適切な使用の推進

これまでに制定された適合性評価に関するISO/IEC規格，ガイドについて表6-4にまとめる。

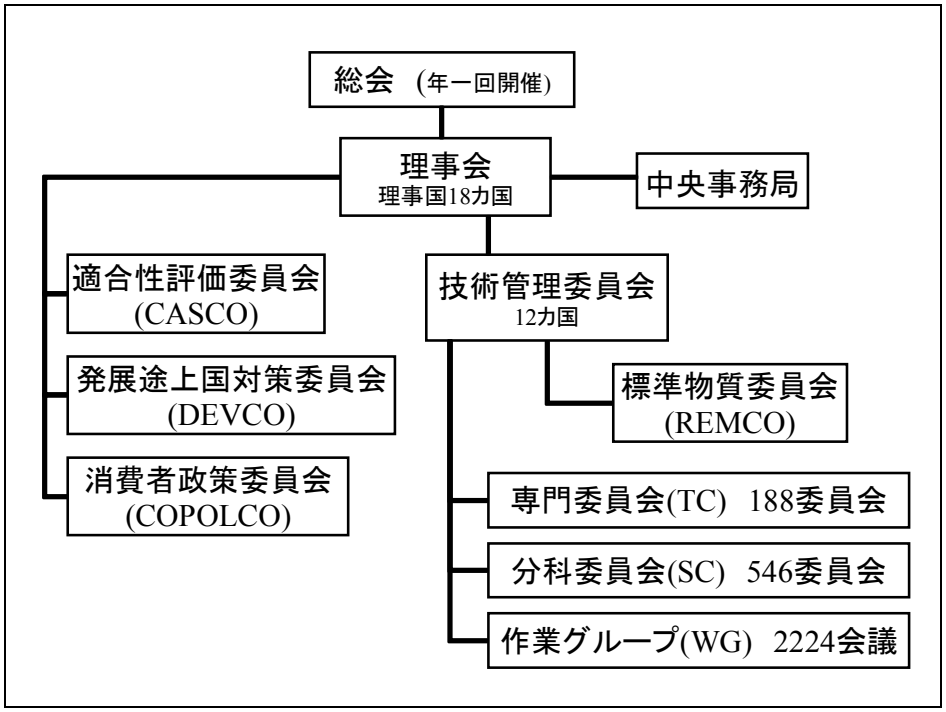


図6-4　国際標準化機構ISO体制図

表6-4　適合性評価に関するISO/IEC規格・ガイド

<table>
<tr><th></th><th>校正及び試験</th><th>検査</th><th colspan="4">認証</th><th>自己適合宣言</th></tr>
<tr><th></th><th></th><th></th><th>製品認証</th><th>QMS認証</th><th>EMS認証</th><th>要員認証</th><th></th></tr>
<tr><td rowspan="2">認定機関に対する要求事項</td><td>ISO/IEC Guide 58</td><td>ISO/IEC TR 17010</td><td>ISO/IEC Guide 61</td><td>ISO/IEC Guide 61</td><td>ISO/IEC Guide 61</td><td>ISO/IEC Guide 61</td><td rowspan="2"></td></tr>
<tr><td colspan="5">ISO/IEC 17011:2004</td></tr>
<tr><td>適合性評価機関に対する要求事項</td><td>ISO/IEC 17025</td><td>ISO/IEC 17020</td><td>ISO/IEC Guide 65</td><td>ISO/IEC Guide 62</td><td>ISO/IEC Guide 66</td><td>ISO/IEC 17024</td><td>ISO/IEC 17050</td></tr>
<tr><td>対象となる組織に対する要求事項</td><td>各種校正・試験方法規格</td><td>各種検査方法規格</td><td>各種製品規格</td><td colspan="2">ISO/IEC 17021</td><td>要員技量試験規格</td><td>各種製品試験方法規格</td></tr>
<tr><td></td><td></td><td></td><td></td><td>ISO9001</td><td>ISO14001</td><td></td><td></td></tr>
</table>

６．２．４　適合性評価体制の具体例

（1）　マネジメントシステム審査登録制度

マネジメントシステム審査登録制度とは，企業，組織における，品質マネジメントシステムQMS(Quality Management System)や環境マネジメントシステムEMS(Environment Management System)がISO9000シリーズやISO14000シリーズなどの規格に適合しているか否かを，購入者などが直接審査する代わりに，第三者機関であるマネジメントシステム審査登録機関が審査し，適合した場合に当該企業組織を登録・公表する制度である。

日本におけるマネジメントシステムの認定機関は，1993年に現在の日本経済団体連合会の傘下35団体の支援を受けて，民間の非営利機関として設立された日本適合性認定委員会JAB(The Japan Accreditation Board for Conformity Assessment)である。

図2.5に示すように，マネジメントシステム審査登録機関自身も第三者適合性評価活動の仕組みの中で，認定機関が認定審査をもとに認定するという仕組みとなっている。

さらにマネジメントシステム審査登録機関に所属する審査員育成のための審査員研修機関・審査員評価登録機関も同様に認定機関による審査を受ける[10]。

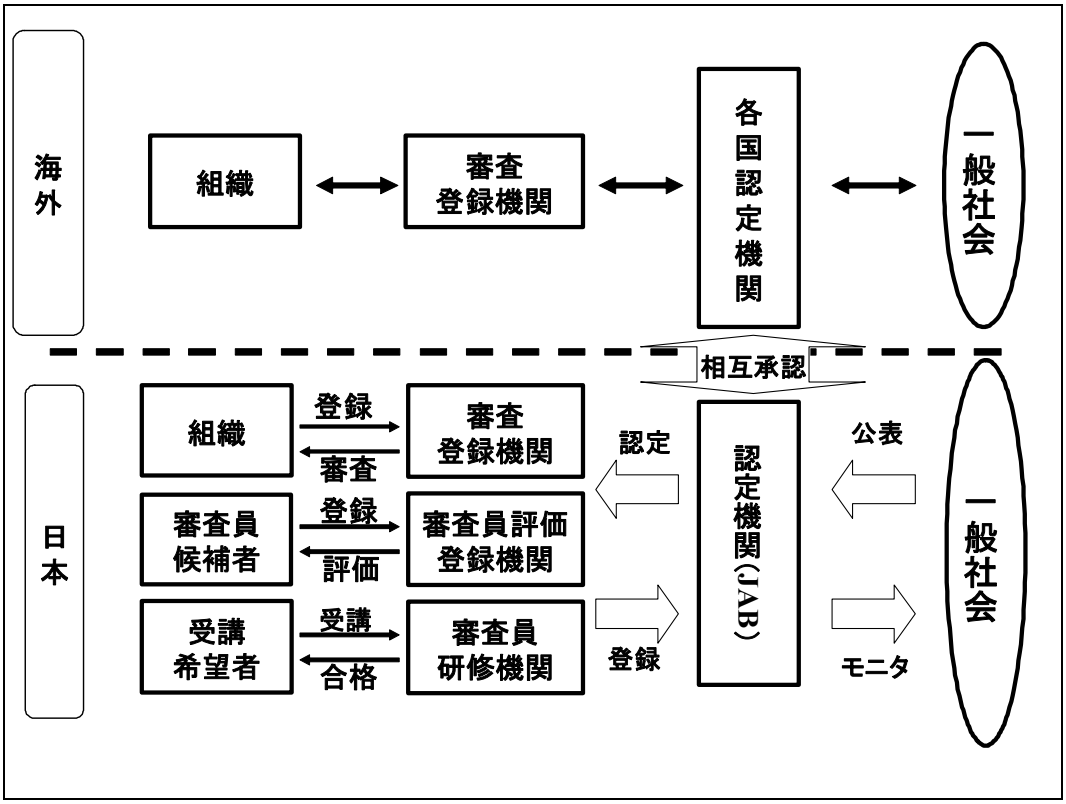


図6-5　マネジメントシステム審査登録制度に対応する日本の体制

（2）　ISO/IEC 15408

適合性評価制度の具体例として，IT関連製品や，特定サイトの情報システムに対するセキュリティを評価，認証のため基準，コモンクライテリアCC（Common Criteria）をベースに作成された国際標準規格，ISO/IEC15408「 Evaluation criteria for IT security(セキュリティ技術 ー情報技術セキュリティの評価基準)」について述べる。

2001年4月，経済産業省は，ITセキュリティ評価及び認証制度 JISEC（Japan Information Technology Security Evaluation and Certification Scheme）を創設し，図6-6に示す第三者評価機関によるIT関連製品のセキュリティ機能，品質の認証制度を整備した。

2003年10月には，CCに基づく認証結果を相互に承認する国際協定である，コモンクライテリア承認アレンジメントCCRA（Common Criteria Recognition　Arrangement）に正式加盟し情報セキュリティ分野の国際的な適合性評価制度を整えている[11]。

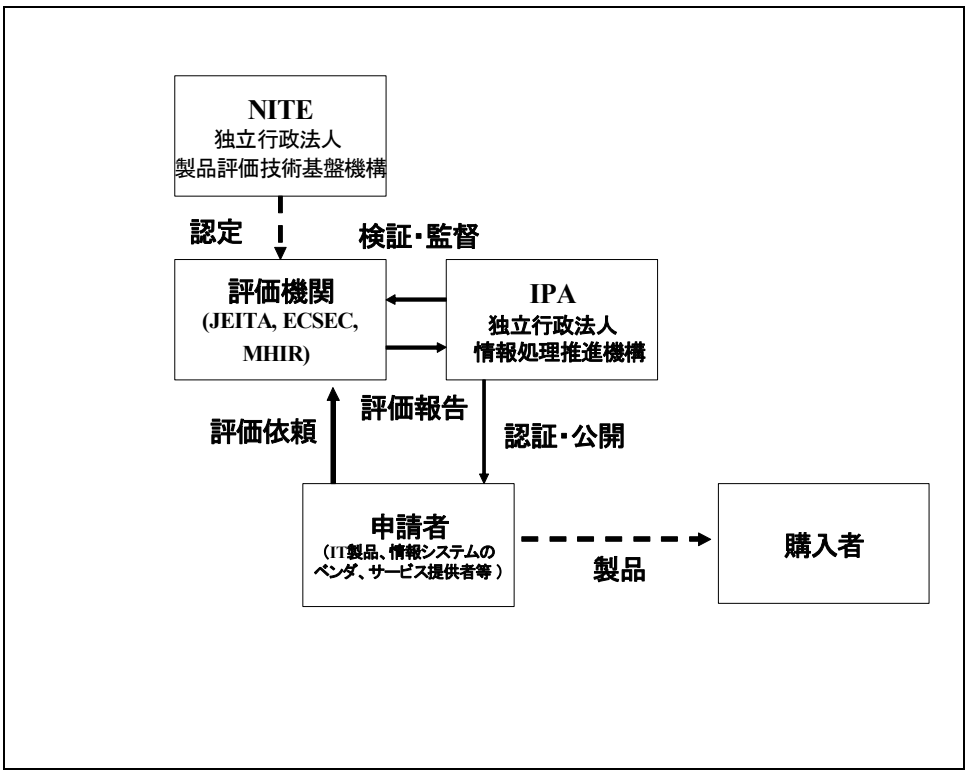


図6-6　ISO15408に対応する日本の体制

６．２．５　まとめ

⑴ 適合性評価の仕組みは，供給者が適合性評価主体となる第一者評価活動，購入者が主体となる第二者評価活動に変わって，国際標準に準拠した認定機関から認証を受けた第三者機関が主体となる仕組みへと変化してきた。また，国ごとに定められ，法的な強制力を持つ強制規格も国家間での共通化が進んでいる。この背景には，経済活動のグローバル化や，EUによる域内統合という流れを経て，WTOにより国際貿易の障壁を最小限にするための国際協定であるWTO/TBT協定が成立したことによる影響が存在する。

⑵ WTO/TBT協定により一民間機関であったISOが，WTOの国際条約としての強制力から標準規格の世界基準として存在感を強め，適合性評価においてはISO/CASCOが発行する規格が，世界基準として重要性を増している。

⑶ 第三者適合性評価活動の枠組みは，例として取り上げたマネジメントシステムや，情報セキュ

リティシステムの標準であるISO/IEC15408に関する組織体制を見てもわかるように，製品認証，システム認証などさまざまな分野で利用されるようになっている。

(4) 日本国内の体制を検討する場合，第一者適合性評価活動，第二者適合性評価活動，第三者適合性評価活動のうち，どの体制を選択するかは対象の性質によるが，国際標準のフレームワークに準拠することが必要と考える。

参考文献

[1] 日本規格協会JIS Q17000 2005.7

[2] 日本工業標準調査会JISC 適合性評価について http://www.jisc.go.jp/acc/index.html

[3] (財) 日本適合性認定協会編： 適合性評価ハンドブック 日科技連 2002

[4] 日本機械工業連合会JMF ＥＵの基準認証制度の問題点
http://www.jmf.or.jp/japanese/wold_topic/17EU_kijun/euki_1_1.html

[5] 経済産業省　産業技術環境局相互承認について
http://www.meti.go.jp/policy/conformity/mrarenew/MR.htm

[6] JISC: JISC関係用語と略語集 http://www.jisc.go.jp/dictionary/index.html#ALM01

[7] 経済産業省 基準認証ユニット 相互承認推進室: 相互承認協定を巡る動向
http://www.jisc.go.jp/jisc/pdf/tekigousei_20pdf/20-7.pdf

[8]奈良好啓：国際標準化入門、日本規格協会　2004

[9] 小田部譲: 標準化教育プログラム 7章　国際規格と国際標準化機関 2007.10

[10] 小田部譲: 標準化教育プログラム 11章　適合性評価認証制度 2007.6

[11] (独) 情報処理推進機構：日立製作所，産業技術大学院大学　受託　ITセキュリティ評価・認証に関するe-learning用教材 2007.9

適合性評価に関しては，下記以外にあわせて20数件の規格・ガイドが発行されている。

付表　適合性評価に関する代表的なISO/IEC規格・ガイド

ISO/CASCO文書名	概要
ISO/IEC Guide 60:2004 Conformity assessment Code of good practice JIS Q0060	「適合性評価－適正実施基準」 適合性評価の基準文書，機関，システム，スキーム及び結果を含め，適合性評価のすべての要素に対して適正実施に関する原則を規定。
ISO/IEC 17000:2004 Conformity assessment Vocabulary and general principles JIS　Q17000:2005	一般用語定義・適合性評価の利用に関する一般用語定義の基準
ISO/IEC 17011:2004 Conformity Assessment General requirements for accreditation bodies accrediting conformity assessment bodies JIS Q17011: 2005	「適合性評価－適合性評価機関の認定を行う認定機関に対する一般要求事項」 適合性評価機関を審査し認定する認定機関に対する一般要求事項を規定。この規格において適合性評価機関とは，試験所，校正機関，検査機関，マネジメントシステム認証機関，要員認証機関及び製品認証機関である。
ISO/IEC 0017040:24 Conformity Assessment General requirements for peer assessment of conformity assessment bodies and accreditation bodies JIS Q17040	「適合性評価－適合性評価機関及び認定機関間の同等性評価に対する一般要求事項」 認定機関または適合性評価機関の合意グループが実施する同等性評価プロセス及びこれに関係する合意グループの構成及び運営に対する一般要求事項を規定。
ISO/IEC 17021 Conformity assessment Requirements for bodies providing audit and certification of management systems 対応JISなし	「適合性評価－マネジメントシステムの監査及び認証を提供する機関に対する要求事項」
ISO/IEC 17024:2003 Conformity Assessment General requirements for bodies operating certification of persons JIS Q17024:2004	「適合性評価－要員の認証を実施する機関に対する一般要求事項」 要員の認証スキームの開発及び維持を含む，特定の要求事項に照らして要員を認証する機関に対する一般要求事項を規定。
ISO/IEC Guide 65:1996 General requirements for bodies operating product certification systems	「製品認証機関に対する一般要求事項」

JIS Q 0065：1997	製品認証業務を行っている第三者機関が，適格であり信頼できると認められるために遵守しなければならない一般要求事項を規定。
ISO/IEC 17025:2005	「試験所及び校正機関の能力に関する一般要求事項」 サンプリングを含め，試験または校正を行う機関の能力に関する一般要求事項を規定。
General requirements for the competence of testing and calibration laboratories	
JIS Q 17025	

６．３　各国の適合性評価制度調査

６．３．１　韓国

（1）　背景

１）　組織および活動

韓国においてバイオメトリック技術開発や標準化推進を行う組織は，表6-5に示すように韓国情報保護振興院KISA(Korea Information Security Agency)を始め，韓国バイオメトリクス協会KBA (Korea Biometrics Association)，韓国電子通信研究所ETRI(Electronics and Telecommunications Research Institute)，韓国情報通信技術協会TTA(Telecommunications Technology Association)，バイオメトリックエンジニアリング研究センターBERC(Biometric Engineering Research Center)がある。

韓国では，情報通信省 MIC(The Ministry of Information and Communication)傘下の研究機関であるKISA主導の元，数々の政策や海外連携が行われている。国内政策では，システム評価，データベース構築，法制度に関する諮問委員会を設立している。特に2002年より開始されたデータベース構築プロジェクトでは，指紋および顔データベースをMICの資金支援により構築し，無償公開することで韓国内でのバイオメトリクス製品開発を支援している。

またシステム評価関連では，BioAPI適合性試験に取組み，ISO/IEC JTC1 SC37においてはWG2(Working Group 2)の国際主査を務めているほか，WG5で策定中の精度評価の標準SO/IEC19795のコエディタを引き受けている。海外の関連機関との連携窓口もKISA が担っており，欧州の団体であるヨーロピアンバイオメトリクスフォーラム　EBF(European Biometrics Forum)との連携も早期から実施している[1]。また，後述するがKISA/K-NBTCはAsia Biometric Consortiumの事務局を担当している。

2008年現在，産業資源部技術標準院主導JEMAの元で，情報関連技術の内，生体認証分野，セキュリティ技術分野を含む，国際市場をリードできると見込まれる15分野205件が国際標準として採択されることを目指す「IT標準化5カ年計画」(2006～2010)を推進している。この計画に基づき，2006年から企業や研究所などで本格的な研究開発が行われ，5年間で合計30億5000万ウォン(約3億500万円)の投資が行われている。

産業資源部によると，これまでISO/IEC JTC1などにより制定されたIT分野の国際標準案件計2334種のうち，0.9%にあたる22種において韓国の技術が採択されている。また国際標準として制定作業中である規格に関しても，制定中の53種に対し83件の韓国提案が採用されている[2]。

IT標準化5カ年計画の対象となる15分野の詳細は以下の通りである。

① 現在国際市場を主導できる可能性の高い分野

- 通信およびシステム間の情報交換　(JTC1/SC6)
- 情報機器相互接続　(JTC1/SC25)
- マルチメディア情報の符号化　(JTC1/SC29)
- マルチメディア機器　(IEC/TC100)

② 国際市場および国際標準化が初期段階にあり先駆け効果が高い分野

- 学習・教育・研修のための情報技術(JTC1/SC36)
- 交通情報制御システム　(ISO/TC204)
- 地理情報・ジオマティクス　(ISO/TC211)

・ 医療情報 (ISO/TC215)
・ ユーザインタフェース (JTC1/SC35)
・ 生体認証 (JTC1/SC37)

③消費者保護など国民の日常生活に影響力が高い分野

・ 文字コード (JTC1/SC2)
・ 識別カー ドと関連機器 (JTC1/SC17)
・ セキュリティ技術 (JTC1/SC27)
・ ドキュメンテーション (ISO/TC46)
・ 金融サービス (ISO/TC68)

表6-5 韓国の国際標準に関わる主な組織

組織	内容
韓国情報保護振興院 KISA (Korea Information Security Agency)	1995年8月に制定された情報化促進基本法に基づいて，1996年4月に設立された。韓国情報通信部MICの関連研究所にあたる。情報セキュリティ分野に関わる技術開発，教育，産業界支援などを行っている。バイオメトリクスの研究は，ITセキュリティ評価&産業支援部門の評価&産業プランニングチームが担当している。
韓国バイオメトリクス協会 KBA (Korea Biometrics Association)	韓国情報通信部MICのバイオメトリクス産業育成の一環として2001年2月に設立された。バイオメトリクス産業および市場の活性化，産官学の協同作業の促進，国際競争力の強化，標準や性能評価，相互運用性に関する情報の提供を目的とする。
韓国電子通信研究所 ETRI (Electronics and Telecommunications Research Institute)	1976年に設立された政府出資の非営利研究機関である。ソフトウェアからネットワーク技術までIT技術に関わる研究を行っている。バイオメトリクスの研究はETRIの情報セキュリティ研究部門のバイオメトリクス技術調査チームで行っている。
韓国情報通信技術協会 TTA(Telecommunications Technology Association)	電気通信標準化や無線通信標準化など，韓国IT技術標準案を審議，認定する機関である。バイオメトリクス関連の標準は，TC10(Technical Committee 10)「情報セキュリティ技術委員会」のSG3(Sub Group 3)「システムとネットワークセキュリティスタディグループ」において審議される。
バイオメトリクスエンジニアリング研究センター BERC (Biometric Engineering Research Center)	バイオメトリクスの研究と研究者育成のため，2002年8月に韓国科学技術部MOST(Ministry of Science and Technology)と韓国科学財団KOSEF(Korea Science and Engineering Foundation)の出資金を基に延世大学内に設立。指紋，顔，虹彩，静脈，音声における最新技術の研究やマルチモーダルの研究に取り組んでいる。

2） 指紋登録制度

韓国では，「住民登録証」または「運転免許証」を発行される全国民が指紋登録することが義

務付けられている。成人(満17歳)に達して住民登録証を取得する手続きを行う際に，10指すべての指紋を採取される。採取された指紋データは韓国内務省MOGAHA(The Ministry of Government and Home Affairs)と警察庁NPA（National Police Agency）のデータベースで管理されている。

この制度は，1962年に制定された「住民登録法」に起因する。住民登録には，住民の居住状況の把握，人口統計の明確化という目的の他に，軍事政権下の住民統制や北朝鮮のスパイを探し出すといった目的もあり，住民登録の際の指紋採取が義務化され，現在も続いている。

住民登録証に記載されている住民登録番号(12桁)は，生年月日，性別，出生地出生順といった，個人情報から構成されている。また，住民登録証を発行する際に捺印した10指の指紋は，その特有の模様によって分類され，個人毎に指紋番号が付与される。民間で住民登録番号を使用することができ，住民登録番号を入手できれば，関連機関を通じてその人の信用情報，財産情報等の全ての個人情報を調べることができる[1]。

（2） 韓国の適合性評価機関K-NBTC

K-NBTCの体制調査については，2007年8月に日本に招聘して行ったK-NBTC所長Jason Kim博士のプレゼンテーション資料KISA at a glanceおよび，アジアバイオメトリクスコンソーシアムABC(Asian Biometrics Consortium)におけるK-NBTC 研究員Youngjun kim のプレゼンテーション資料Schemes and Procedures of K-NBTC for Conformance Test(CTS Service of K-NBTC)を参考とした。K-NBTCの試験内容詳細についてはセキュアデザイン研究所の栗田寛久にヒアリング調査を行った。

1） 韓国国立バイオメトリックテストセンターK-NBTC概要

韓国国立バイオメトリックテストセンターK-NBTC（Korea National biometric Test Center）は2005年4月の情報通信省MICの委員会における提起，総合的なバイオメトリクス基盤設立のための計画（Plan for Establishment on Overall Infrastructure for Biometrics）に基づいて2006年6月に設立された。

K-NBTCの主なスコープは以下の通りである。

①指紋認証，虹彩認証アルゴリズムパフォーマンス測定テストのための機器やBioAPI2。0への適合性評価テスト用機器，相互運用性テスト用機器などの各種テスト用プロトコルの開発

②SC37-KOREAを通じての国際標準に関する活動

③国内の標準化活動，および調査活動

④KISAと連携してのアジアバイオメトリックコンソーシアムABC(Asian Biometric Consortium)やヨーロッピアンバイオメトリクスフォーラムEBF(European Biometrics Forum)，英国バイオメトリックワーキンググループUKBWG(UK Biometric Working Group)などとの国際協業への貢献

⑤これらの活動を通じての韓国工業の活性化および，国際競争力の強化への貢献

上記の他，2007年5月よりAsia Biometric Consortiumの事務局設置および公式ホームページの開設をK-NBTCが担うことになった。

2007年の活動は，1月から企業へのコンサルティングサービスの提供，4月末にシナリオテスト体制の整備，7月からBioAPIに準拠した適合性，顔，虹彩，静脈認証アルゴリズムのパフォー

マンス測定テストを行っている。

K-NBTC所長であるJason Kim博士へのヒアリングによると，2007年9月現在，組織の運営体制は総勢6名で担っている。経済的な費用は組織の運営・企業の試験受験費用を含め，全て政府が出資している。今後認定制度が普及していけば民営化も考えられているという。生体認証に関わる政府調達品については，調達の基準がK-NBTCによる認証を受けた製品とされた。これにより韓国国内で生体認証製品に関わる入札参加にはK-NBTCによる認証が必要となるような方向で動いている。また他国への輸出製品についての適合性試験も行っている。

K-NBTCのサイト上には，表6-6に示すようにDigent社の製品Izzix-BSP V1・Izzix-BSP V2および，Nitgen社のNitgen BSP for BioAPI2.0について適合性評価試験の試験記録が公表されている。

K-NBTCの活動に関係する組織として，図6-8の機関が挙げられている。国内機関としては，すでに3.1.1.1で説明した各機関の他にICカードリサーチセンターがある。海外の組織としては，標準化活動としてISO/IEC JTC1/SC37に加えて，国際電気通信連合 電気通信標準化部門ITU-TのSG17へも参加している。

表6-6　K-NBTCサイト上に公表されている適合性評価認定製品

Tested Products

KISA Home > KNBTC > Tested Products > **BioAPI Conformance**

Product Name	Manufacturer	Conformance Claim	Valid. Date
Izzix-BSP V1(Verification)	Digent	Conformity	
Izzix-BSP V2.2(Verification)	Digent	Conformity	
NITGEN BSP for BioAPI 2.0	Nitgen	Conformity	

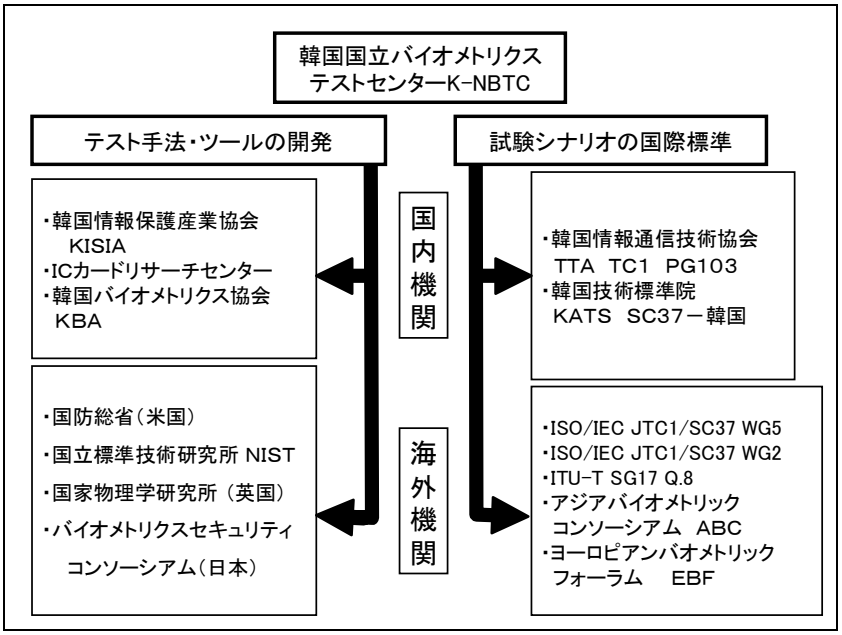


図6-8　K-NBTCを中心にみた韓国の体制

2）　K-NBTCが提供するサービスの概要

K-NBTCが2007年11月時点で提供しているサービスを以下にまとめる。

①韓国内のバイオメトリックサービスベンダー の製品に対するテスト・認証サービス

①BioAPI適合性評価（ISO BioAPI Ver2.0）　2006年7月開始

②アルゴリズム性能評価（指紋，顔）　2006年10月開始

③相互運用性テスト（指紋データ）　2006年12月開始

②韓国内のバイオメトリクス製品ベンダーの製品に対するテスト・認証サービス

①BioAPI適合性評価（ISO BioAPI Ver2.0）　2006年7月開始

②アルゴリズム性能評価（指紋，顔）　2006年10月開始

③相互運用性テスト（指紋データ）　2006年12月開始

③生体認証分野での政府主導プロジェクトの技術的援助，サポート

国内公共機関の生体認証研究に対する技術的な援助やサポート

④国内外の生体認証研究・開発機関との連携

図3.2はK-NBTCの設備図のコピーである。この図から以下のことが推測できる。

試験用入力端末は指紋センサー，虹彩センサー，顔画像センサー，静脈センサーを備えている。試験システムとして相互運用性試験システム，ANSIとISOの2つの基準に対応したBioAPI適合性評価試験システム，アルゴリズム性能評価システムを備えている。またシナリオテスト用の顔画像用2次元スタジオ，3次元スキャナや気温，湿度などへの対応を試験する試験室も備えている。

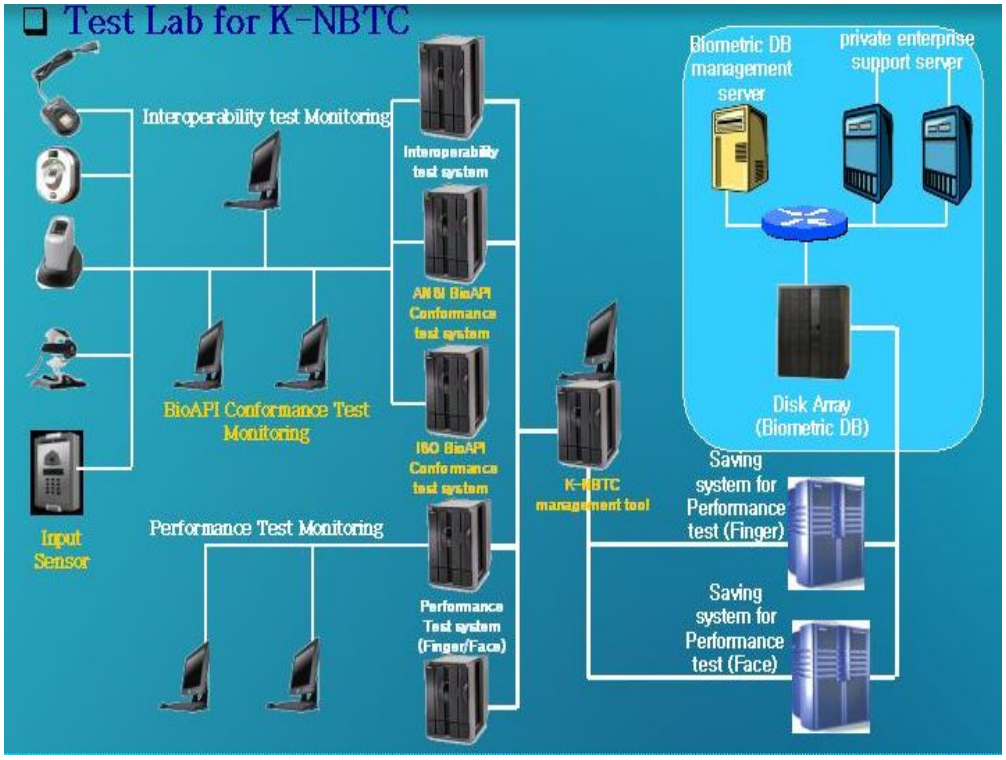


図6-9　K-NBTCテスト環境

3）　K-NBTCのテスト・認証サービス

K-NBTCのテスト・認証体制について，テストの流れと手順についてまとめる。K-NBTCにおける適合性評価テストの開始から終了までの流れは，図6-10のように大きく3段階に分かれる。試験に要する期間についての情報はJason Kim博士からのヒアリングに基づいている。

①試験前段階：　手続きに4-8週間程度

- 適合性試験希望者は，試験用製品を準備し試験に申し込む。
- テスト機関は申し込みを受理し契約を行い，試験内容ついての相談に応じる。
- 認定機関は，テスト機関が交わした契約内容と試験製品の確認を行う。

②試験段階：　手続きに2-4週間程度

- 適合性試験希望者は，試験用製品を提供する。試験を中断することも可能である。
- テスト機関は，試験チームを組織し試験を行い，その結果をレポートとしてまとめる。
- 認定機関は，テスト機関から提出された試験計画を認可する。

③試験終了段階：　手続きに4週間程度

- 適合性試験申込者は，試験結果と製品への認証を得る。
- テスト機関は，申込者に試験結果を通知する。
- 認定機関は。結果レポートと認証を申込者へと発行し，認証された製品を認証台帳に登録する。

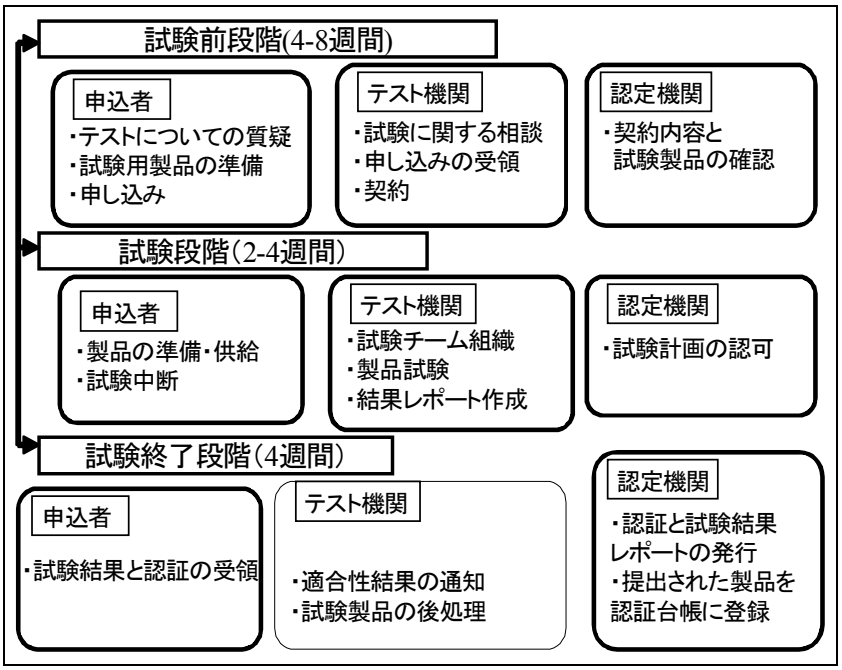


図6-10　適合性評価テストの流れ

(A)　BioAPI2.0適合性評価試験

BioAPI適合性試験の具体的な内容については，セキュアデザイン研究所　栗田寛久　へのヒアリングに基づいている。

BioAPI試験は，バイオメトリクス製品がISO/IEC JTC1/SC37で定められたBioAPIVer2.0に適合しているかどうかを調べることを目的とした試験である。試験の概要や結果のフォーマットなどについてはISO/IEC 24709－1として国際標準になっている。

BioAPI試験の流れは図6-11に示すように

A)　BioAPIモジュールの起動
B)　GUIによる機能選択
C)　製品機能正確性試験*
D) 機能試験（Function Test）

試験の方針は反証テストである。具体的には試験用ソフトウェアから表6-7にあげたBioAPI関数の呼び出しを行い，その成否によって適合性を判定する。

という流れである。

*試験製品がメーカーの公表している仕様を満たしているかを確認する準備試験

表6-7　BioAPIが提供する主要関数

種別	機能	APIの関数名	SPIの関数名
バイオメトリクス関数	登録	BioAPI_Enroll	BioSPI_Enroll
	照合	BioAPI_Verify	BioSPI_Verify
	識別	BioAPI_Identify	BioSPI_Identify
	生体情報の取得	BioAPI_Capture	BioSPI_Capture
	テンプレート作成	BioAPI_Create template	BioSPI_Create template
	生体情報の処理	BioAPI_Process	BioSPI_Process
	マッチング	BioAPI_Verify Match	BioSPI_Verify Match
データベース関数	BIRの追加	BioAPI_DbstoreBIR	BioSPI_DbstoreBIR
	BIRの読込	BioAPI_DbGetBIR	BioSPI_DbGetBIR
	BIRの照会	BioAPI_DbQueryDIR	BioSPI_DbQueryDIR
	BIRの削除	BioAPI_DbDeleteDIR	BioSPI_DbDeleteDIR
フレームワーク関数	初期化	BioAPI_Init	BioSPI_Init
	BSPのロード	BioAPI_ModuleLord	BioSPI_ModuleLord
	BSPのアンロード	BioAPI_ModuleUnlord	BioSPI_ModuleUnlord

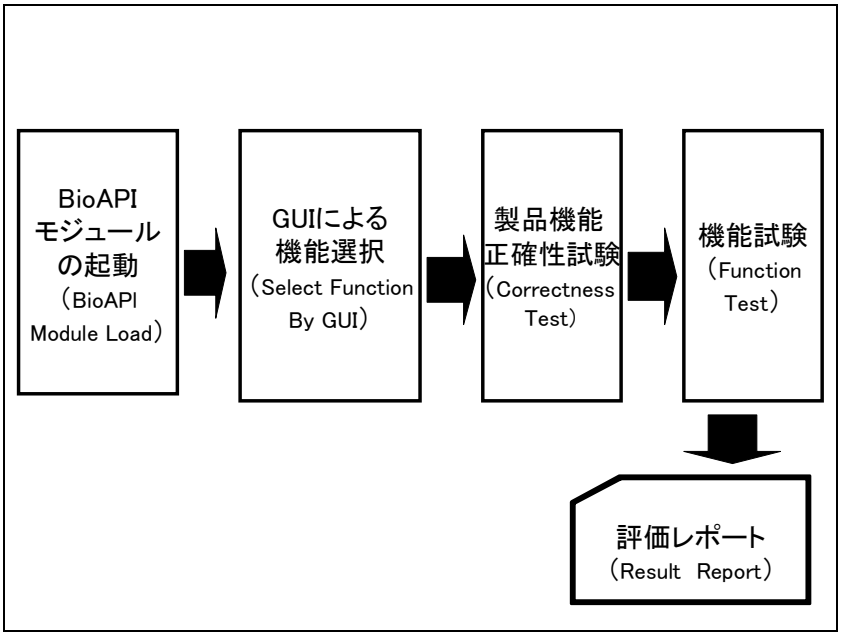


図6-11　BioAPI適合性評価試験の流れ

表6-8　BioAPI適合性評価試験結果画面サンプル

Functions		Result (A)	Result (K-NBTC)	
Component Management Functions				
	BioSPI_BSPLoad	No Problem	PASS	
	BioSPI_BSPUnload	No Problem	PASS	
	BioSPI_BSPAttach	No Problem	PASS	
	BioSPI_BSPDetach	No Problem	PASS	
	BioSPI_QueryUnits	No Problem	PASS	
	BioSPI_ControlUnit	No Problem	PASS	BSP, Capture
Handle Functions				
	BioSPI_FreeBIRHandle	No Problem	PASS	
	BioSPI_GetBIRFromHandle	No Problem	PASS	
	BioSPI_GetHeaderFromHandle	No Problem	PASS	
Callback and Event Functions				
	BioSPI_EnableEvents	No Problem	PASS	

表6-8，図6-18はK-NBTCのBioAPI試験に関する結果画面，および画面遷移サンプルである。

注）反証テスト：　BioAPIに適合するのかを決めるためのテストポリシーである。本来BioAPIに適合していることを証明するには，考えられるあらゆる可能性を試験しなければならないが，コストの制約上不可能でである。したがって，何をもって適合しているかは，反証可能性(ISOで決められたBioAPIが備えていなければいけない関数を外部から呼び出しても，うまく反応しない)を実験によって反証されていないことを観察したこと(関数呼び出しが成功する)をもって適合していると考える。

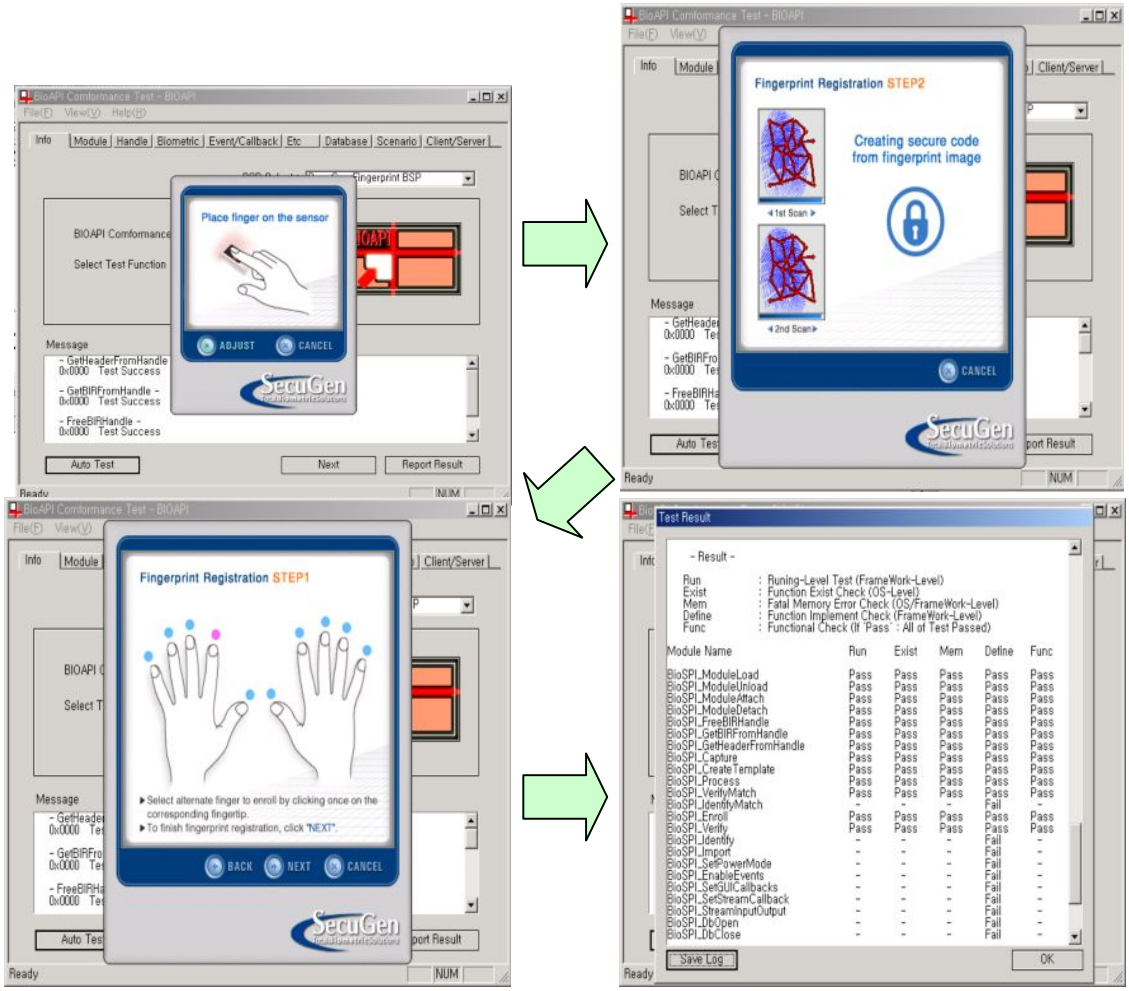


図6-12　BioAPI適合性評価試験画面遷移サンプル

(B)　アルゴリズム精度評価試験

図6-13に示すようにバイオメトリックシステムは，大きく入力，照合，判定の3つの機能に分割できる。このうち測定の対象になるのは照合フェーズである。例として指紋照合アルゴリズムをあげると，大きくはマニューシャ方式，マニューシャリレーション方式，イメージマッチング方式の3種類がある。アルゴリズム精度評価試験では特徴点抽出機能やパターンマッチング機能の精度を試験によって測定する[3]。

図6-14はアルゴリズム精度評価試験の流れである。この試験では，特徴点抽出アルゴリズムモジュール，アルゴリズム照合モジュール，パフォーマンス測定モジュールといったモジュール名からもわかるように，指紋データ，および顔画像データの読取アルゴリズムの精度や読取データとデータベースに格納されているデータとの照合アルゴリズムの精度の測定を行う。

アルゴリズム精度評価試験とシナリオテストに関してはパフォーマンステストとして国際標準ISO/IEC 19795-1で試験項目が定義されている。　図6-15，図6-16，および図6-17は，K-NBTCで

のアルゴリズム性能評価試験に関する結果画面および，画面遷移サンプルである[4]。

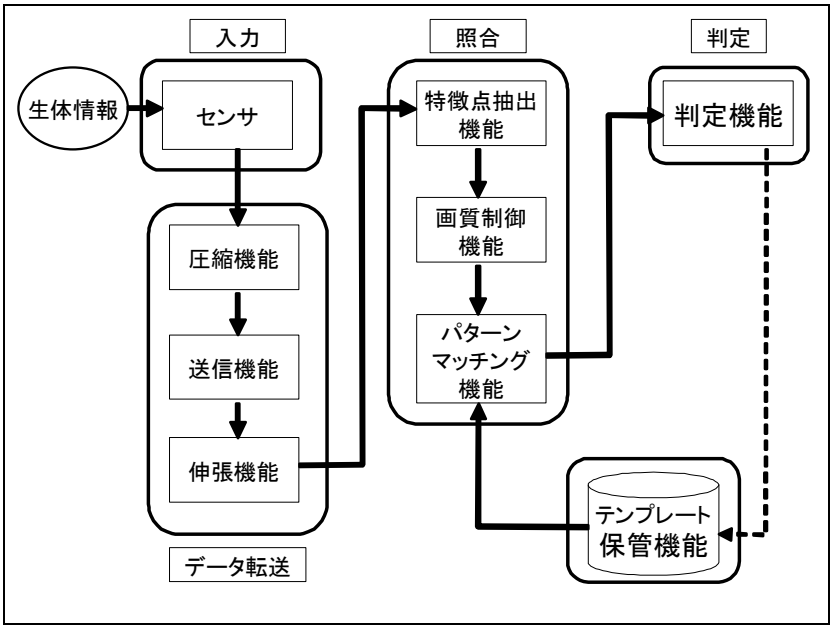


図6-13　一般的なバイオメトリクスシステムの認証構成

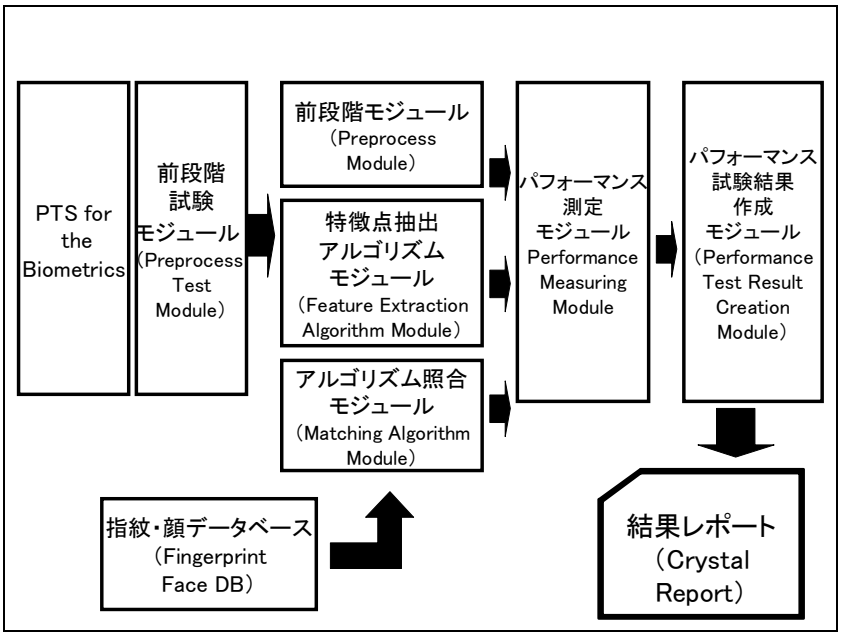


図6-14　アルゴリズム精度評価試験の流れ

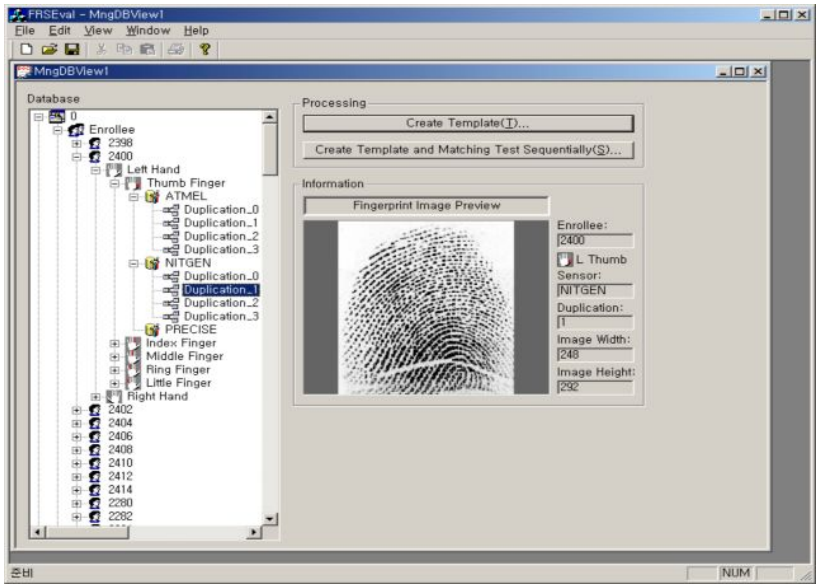


図6-15　アルゴリズム精度評価試験画面サンプル1

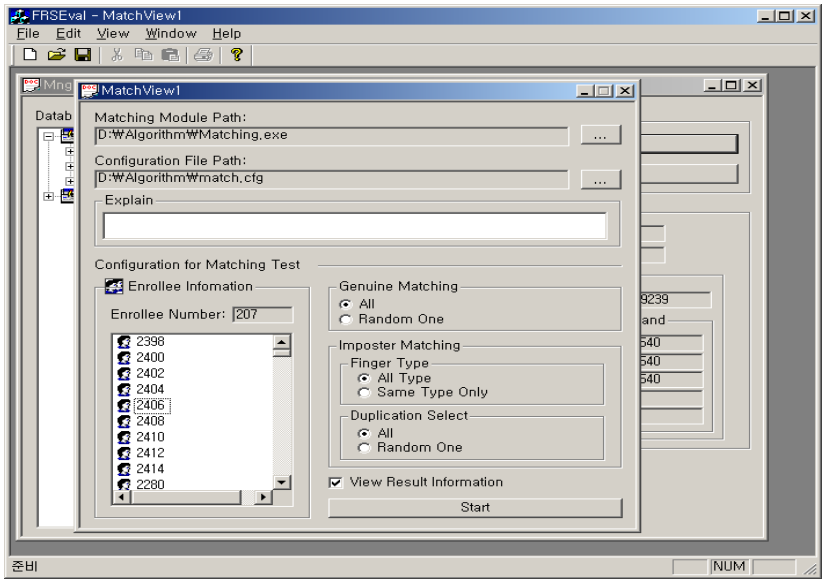


図6-16　アルゴリズム精度評価試験画面サンプル2

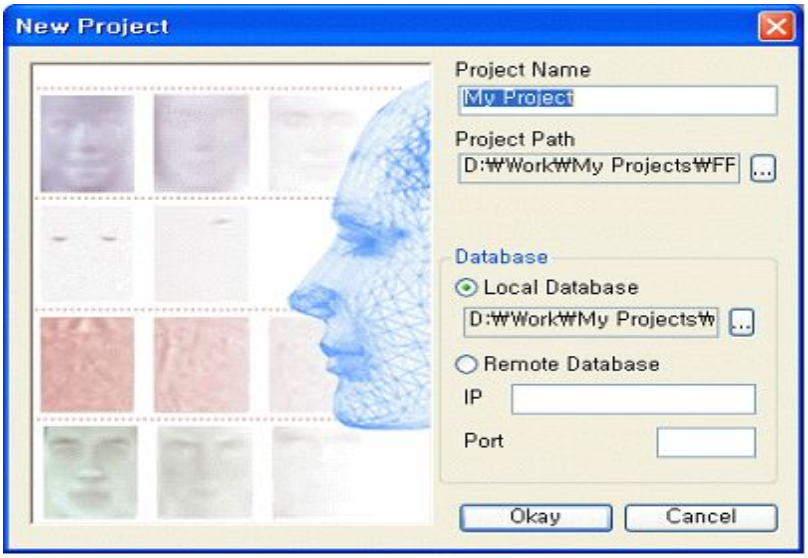


図6-17　アルゴリズム精度評価試験画面サンプル3

(C)　相互運用性試験

図6-18はK-NBTCが相互運用性試験の例として示しているものである。

B社の指紋センサーを採用しているA銀行で，生体認証データベースに登録された指紋データが，D社の指紋照合システムでも同じように認識されるか，またC社の指紋センサーを利用して抽出した指紋テンプレートデータを，D社の指紋照合システムで利用できるかといった，CBEFF規格への適合性と相互利用性を試験によって検証する。図6-19，図6-20は，相互運用性試験に関する画面サンプルである[4]。

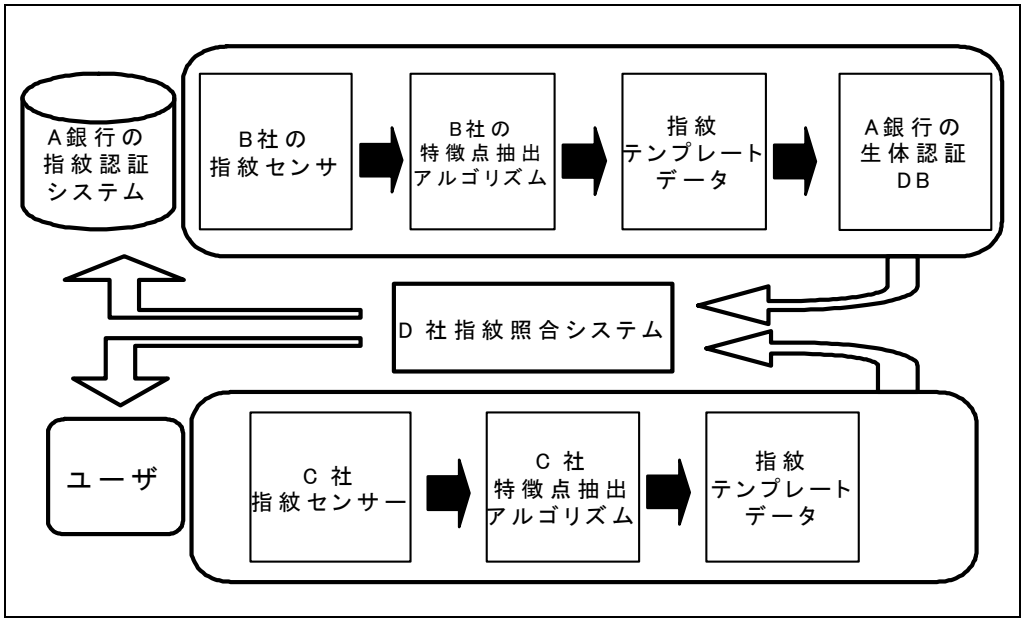


図6-18　相互運用性試験の流れ（指紋データ）

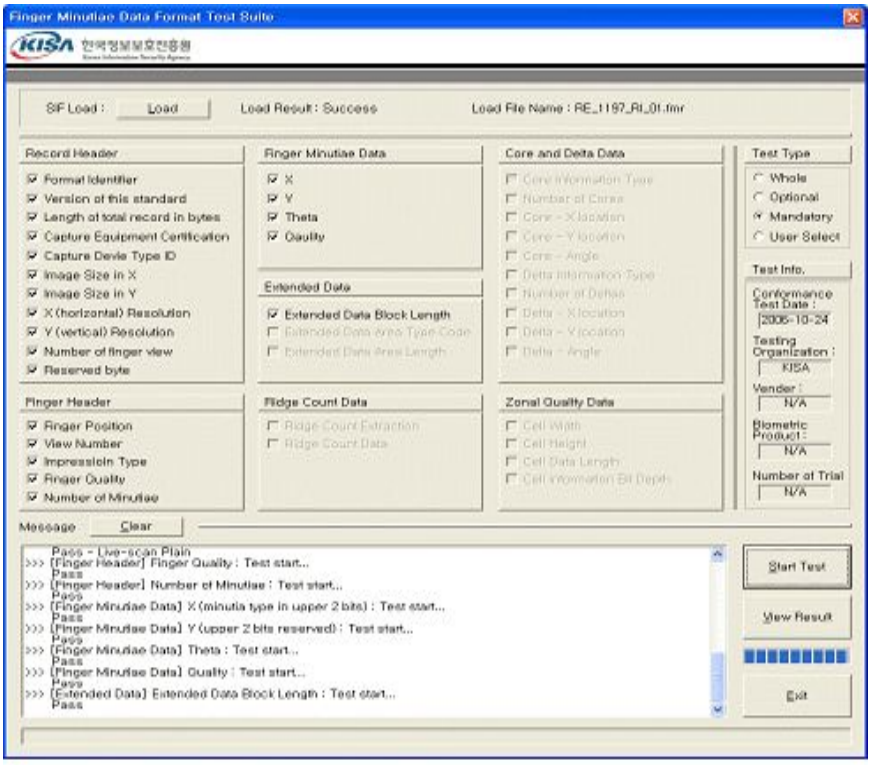


図6-19　相互運用性試験システム画面サンプル1

図6-20　相互運用性試験システム画面サンプル2

(D)　シナリオテスト

シナリオテストは，指紋や顔画像について気温，湿度，顔画像の証明や角度など実際の運用で想定される様々な状況に対する性能をテストする。

図6-21，図6-22，および図6-23は，シナリオテスト設備に関する画面サンプルである[5]。図6-21は環境変化への対応を試験するための試験室，図6-22が2次元顔画像撮影用スタジオ，図6-23が顔画像撮影用3次元スキャナである。

図6-21　シナリオテスト環境（試験室）

図6-22　シナリオテスト環境（顔画像用2次元スタジオ）

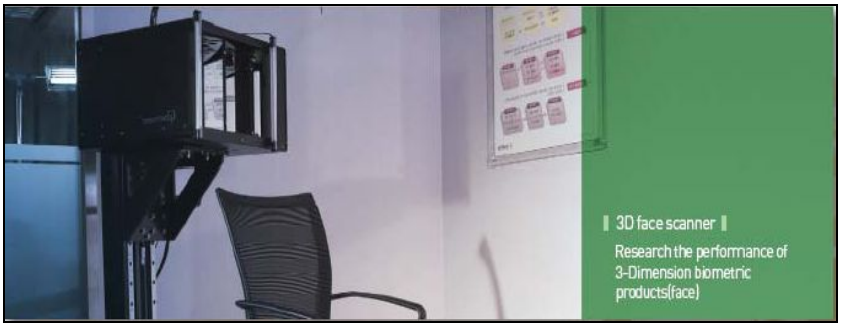


図6-23　シナリオテスト環境（顔画像用3次元スキャナ）

６．３．２　シンガポール

（1）　背景

1）　組織および活動

シンガポールの標準化機関は，シンガポール規格生産性革新庁SPRING(Standards Productivity and Innovation Board)である。SPRINGは生産性と革新，標準と品質，中小企業支援の3分野において活動する。標準化実施の開発促進とともに，国際規格との適合性確保を通じて，シンガポール産の製品・サービスの競争力強化に取り組んでいる。また，適合性評価機関の認定機関，計量標準の供給等の役割を持っており，シンガポールの産業技術政策において重要な役割を担っている。

国内標準化の最高承認機関は，SPRINGによって任命されたStandards Councilである。Standards Councilの委員は，鍵となる分野に所属しその業界をリードできる人物が政府，民間団体，大学等から選ばれる。ここでシンガポール規格関連文書を開発する際の政策やプログラム，手順を決定する。また，国際標準化活動へ参加するための国家標準化プログラムの実施についても方向性を示す。

Standards Councilの下には，10の分野のStandards Committeeが設置されており，さらにその下に分野に応じたTechnical Committeeが設置され，実際の規格開発審議を行っている。

シンガポールの国家規格は，SS(Singapore Standards)規格である。規格番号は1から始まる一連の番号で分野別の分類はない。最近制定された国際規格と整合が図られた規格は，SS ISO・SSIEC等のように，国際規格と番号も含めた整合がなされている。従来から無駄な規格を制定せず，国際規格(ISO,IEC)や英国規格(BS)等を採用し，簡素化を図っていることがシンガポールにおける規格の特徴である。

シンガポールのIT分野の標準化は，ITSC(The Information Technology Standards Committee)が担当している。ITSCは，1990年SPRINGによって任命されたStandards Councilの監督の下に設立された。シンガポールのIT標準化は，貿易産業省の傘下にあるSPRINGではなく，情報通信芸術省傘下の情報通信開発庁によって実施されている。このようにIT分野では，政策と標準化活動が同一の組織で行われるという合理的，効率的な体制が構築されている。

ITSCには，図6-24に示すようにTechnical Committeeとは別に，標準化の利害の代表者から構成されるITSC Council という諮問会議がある。Councilは，ITSC 会長を任命する仕組みとなっており，現在はAccenture社のRobert Chew　が会長を務めている。

現在，350名の専門家と180の機関からの代表者が，ITSCの標準化活動に参画している。このようにして，作成されたIT関連SS規格の数は2004年10月の時点で46規格である[6]。

Technical CommitteeのひとつにBiometricがあり，Yau Wei Yun博士が委員長を務めている。その中で以下のWGが活動している。カッコ内の人物名は各WGの委員長である。ちなみにYau Wei Yun博士はAsia Biometric Consortiumの主要メンバーである。

- Harmonized Biometric Vocabulary WG – (Mr. Arun Kumar)
- Biometric Technical Interfaces WG – (Mr. Goh Jwee Sheng)
- Biometric Data Interchange Formats WG – (Mr. Saurav Bhattacharyya)
- Biometric Functional Architecture and Related Profiles WG – (Mr. Lawrence Chen)
- Biometric Testing and Reporting WG – (Prof. Jiang Xudong)

・ Cross-Jurisdictional and Societal Aspects WG – (Prof. Terence Sim)

生体認証の適合性に研究などで関わっている機関は，この他に2002年に国内の研究開発を促進するために設立された公的機関The Agency for Science, Technology and Research (A*STAR)のグループ機関で，情報，コミュニケーション，メディア(ICM)技術の研究開発を行うInstitute for Infocomm Research（I2R）がある。また組織体制は不明であるがTemasek Polytechnicに複数の研究者がいる。

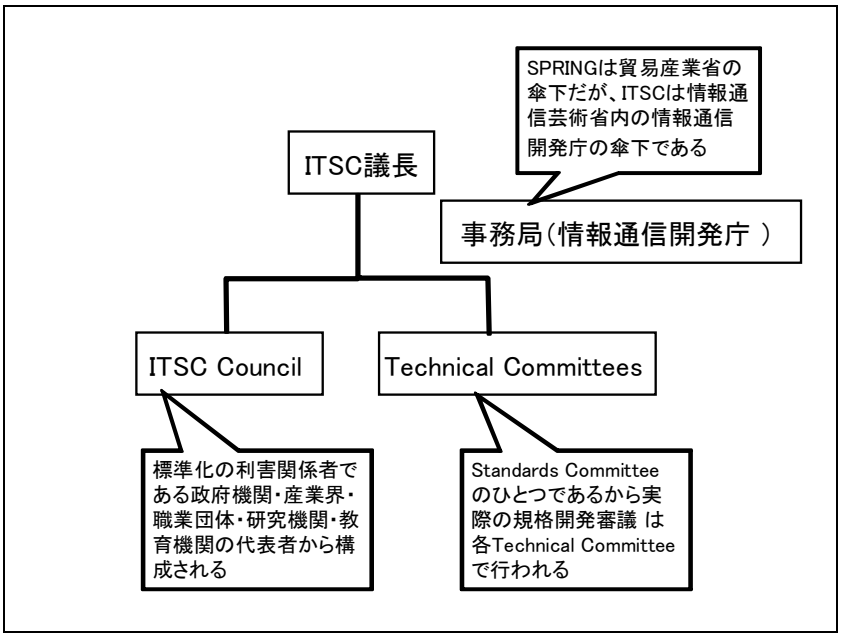


図6-24　ITSC体制図

2）　IT先進国への経緯

シンガポールは，1980年代初頭の国家コンピュータ化計画以降，国家IT計画，IT2000，Infocomm 21を経て，Infocomm21の改訂版であるConnected Singapore，さらにiN2015という一連のIT政策を展開している。またIT産業の誘致，通信インフラの整備，コンピュータ関連法制度の整備，人材育成，電子政府の充実，教育のIT化，通信の自由化等，一連の政策を実施し，ASEANのみならず，世界においても第一級のIT先進国となっている。

1992年に発表されたIT2000計画は，シンガポールをインテリジェントアイランド化するための計画であり，社会生活のあらゆる側面におけるITの採用を推進するものであった。

IT2000の次のマスタープランであるInfocomm21においてはIT産業振興を焦点とし，通信の自由化，情報通信技術の人材育成，情報通信技術産業の振興，電子政府（E-Government行動計画）が進められた。2003年3月に発表されたConnected Singaporeは，新たな政策というよりも，Infocomm21の改訂との位置付けでITを利用する産業，社会を中心に据えたものである。この計画に基づき2003年7月シンガポール財務省および情報通信開発庁は，新たな電子政府計画”eGAPⅡ”を発表した[6]。

2005年3月にはIntelligent Nation 2015(iN2015)の策定が発表された，iN2015は，政府が2015年に向けた10ヵ年計画として策定した計画で，「インテリジェントネイション」あるいは「グローバル

シティ」としてのシンガポールを目指すものである。政府は2015年の目標として以下の6項目を掲げている[7]。

① 情報通信により経済，社会への価値を付加することで世界一を目指す
② 情報通信産業による付加価値を2倍の260億シンガポールドル(約1兆9580億円)にする
③ 情報通信の輸出額を3倍の600億シンガポールドル(約4兆5186億円)にする
④ 8万人の雇用拡大
⑤ 90%の世帯でブロードバンドを利用できるようにする
⑥ 学校へ通う子供のいる全ての世帯でコンピュータを利用できるようにする。

（2） 適合性評価活動

シンガポールは，適合性評価に関する活動は先行しているものの2008年1月時点では，適合性評価活動を専門に行う機関はまだ組織されていない。ただし，2007年9月にシンガポールで開催されたアジアバイオメトリックコンソーシアムカンファレンスABC(Asian Biometric Consortium) Conference 2007（日本の窓口は，（社）日本自動認識システム協会，産業技術大学院大学の瀬戸が実行委員となっている）において，ABCとI2R，Temasek Polytechnic(TP)の3機関が主催，ITSC Biometrics Technical Committee，K-NBTCが共催という形で，企業製品への無料の適合性試験が実施された[8]。

試験内容は，I2R とTemasek Polytechnicが共同開発した適合性評価用ソフトウェアを使用し，Temasek Polytechnicの学生が実試験を行うという体制であった。

表6-9の申し込みフォームのコピーからもわかるように，指紋特徴点データ，指紋画像データ，顔画像データ，データ構造，BioAPIの5種類が用意された。(BioAPI試験に関してはK-NBTC開発のソフトウェアを使用)参加企業はNECの１社のみであった。NECはデータ交換フォーマット、ＣＢＥＦＦフォーマットに関する4つの適合性試験に合格している[9]。

Yau Wei YunとChen Tai Pang, LawrenceがABCのプレゼンテーションで使用した「ISO Standards on General Biometric Technologies & Conformance Testing Institute for Infocomm Research」によると適合性評価用ソフトウェアの開発者としてI2RのYau Wei YunとChen Tai Pang Lawrenceの2名，Temasek PolytechnicのLawrence Lee，Lim Teck Choow，Lim Eyung，Lock Hun Yaの計6名の名前が記載されている[9]。

I2RとTemasek Polytechnicが共同開発した指紋特徴点データ(ISO/IEC 19794-2)，指紋画像データ(ISO/IEC 19794-4)，顔画像データ(ISO/IEC 19794-5)，データ構造(ISO/IEC 19785-1）の4種類の適合性評価用ソフトウェアのうち，ISO Standards on General Biometric Technologies & Conformance Testing Institute for Infocomm Researchから指紋特徴点データ(ISO/IEC 19794-2)に関する試験の流れを図6-25に，また参考資料としてABC2007内で撮影した画像を図6-26および6-27に示す。

表6-9　適合性評価試験申し込みフォーム

Name of Company:	
Contact Person:	
Company Address:	
Contact No:	
Email:	

Please tick the standards that you are interested in participating:		
Tick	Document No.	Type
	ISO/IEX 19794-2	Finger Minutiae Data
	ISO/IEC 19794-4	Finger Image Data
	ISO/IEC 19794-5	Face Image Data
	ISO/IEC 19785-1	Data Element Specification
	ISO/IEC 19784-1	BioAPI specification

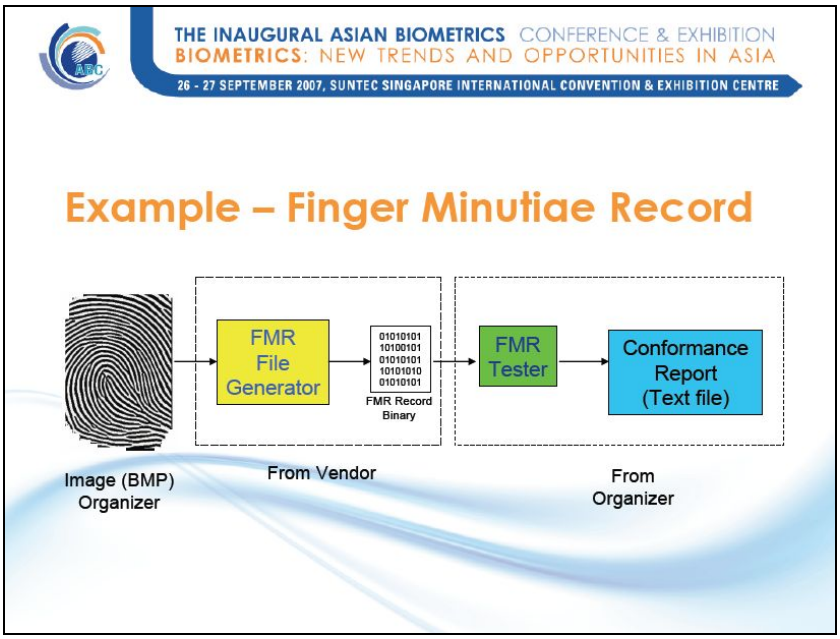

THE INAUGURAL ASIAN BIOMETRICS CONFERENCE & EXHIBITION
BIOMETRICS: NEW TRENDS AND OPPORTUNITIES IN ASIA
26 - 27 SEPTEMBER 2007, SUNTEC SINGAPORE INTERNATIONAL CONVENTION & EXHIBITION CENTRE
Example – Finger Minutiae Record
FMR File Generator
01010101 10100101 01010101 10101010 01010101
FMR Record Binary
FMR Tester
Conformance Report (Text file)
Image (BMP) Organizer
From Vendor
From Organizer

図6-25　指紋特徴点データ評価試験の流れ

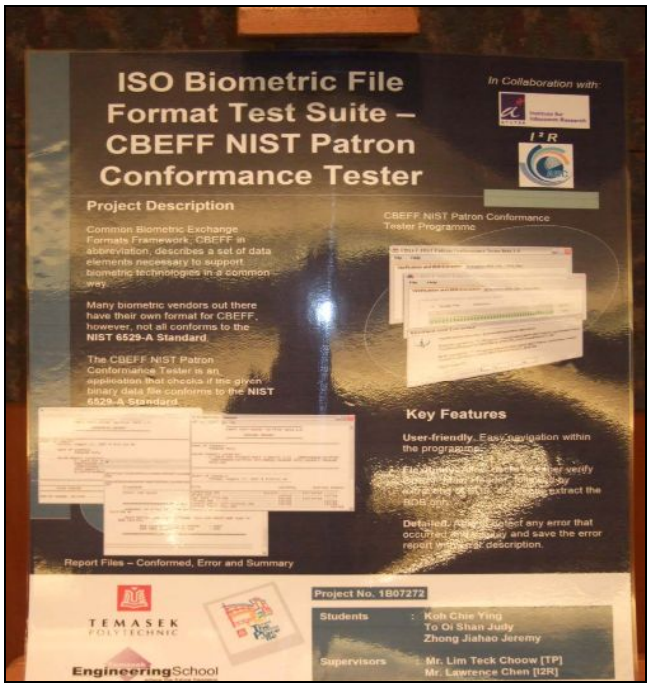


図6-26　適合性試験テスターを説明した展示品

図6-27　指紋認証試験用ソフトウェアを説明した展示品

６．３．３　ＥＵ

（１）　背景

１）　組織および活動

欧州におけるバイオメトリクス検討の特徴としては，技術本位ではなく利用者の立場にたった検討が行われていることにある。欧州ではプライバシーに敏感である背景も加わり，プライバシー保護，関連法規，利用者受容性といった非技術的側面への関心も高い。制度面への関心の高さの一例として，ドイツのバイオメトリクス関連検討組織のリーダーが法律専門家であることをあげることができる。ここでの法律専門家の役割の一つとして，応用案件に対する現行法のもとでの実現可能性の検証がある。このように実行可能性を技術的側面だけでなく制度面からも検討する体制が組まれている[1]。

EUとしての生体認証に関する主要な活動としては，EUの行政執行機関である欧州委員会EC(European Commission)が統括する第4期研究開発プログラムFP4 (Framework Programme)の一環として，1994年から1998年にかけて実施された情報通信分野の調査研究プロジェクトEsprit内の調査，開発プロジェクトBIOTESTがある[11]。

BIOTESTプロジェクトは，1999年12月にバイオメトリクスワーキンググループBWG(Biometric Working Group)により公開され，生体認証の精度評価の国際標準となったBest Practiceバージョン1の元となっている[11]。ちなみに，（株）日立製作所が（独）情報処理推進機構IPAの支援により開発し，現在ＪＩＳ/ＴＳおよびＩＳＯ/ＴＲになった精度評価ガイドラインは，瀬戸洋一（元日立製作所、現産業技術大学院大学）がBWGのメンバーであったため，数学スキームをBest Practiceと整合をとることに成功している。このため，国際標準化において各国の賛同を得ることが容易となった。

2002年6月より約1年間，第5期研究開発プログラム FP5 (Framework Programme 5)の情報社会技術 IST (Information Society Technologies)プロジェクトの一環として実施されたBIOVISIONプロジェクトでは，欧州におけるバイオメトリクス普及のための2003年から2010 年までのロードマップが検討された。また欧州横断的な業界団体となるヨーロピアンバイオメトリックフォーラム EBF（European Biometrics Forum）を設立した。

EBFは，欧州委員会とアイルランド政府から出資を受け2003年7月21日に設立された。本部はアイルランドの首都ダブリンに設置されており，ダブリンに拠点を置くDaon社のMartin Walsh　がチェアマンを務める。

欧州委員会ECは，ヨーロピアンバイオメトリックフォーラム EBFの役割を国際標準技術の確立，バイオメトリクスアプリケーションに対する社会的受容性を高めるための方策や新機軸の検討と位置付けている。ただし，EBFは民間企業であるDaon社の意向が強い運営になっているようであり，EUとしての活動成果がでていないように思える。

また各国ごとにバイオメトリクス研究組織が複数存在する。ここでは主にヨーロピアンバイオメトリックフォーラム EBFと関係がある代表的な組織について表6-10にまとめる[10]。

表6-10　EU域内の主要なバイオメトリクス関連組織

バイオメトリクス ワーキンググループ BWG(Biometric Working Group) (英国)	BWGは，英国政府通信本部GCHQ (Government Communication Headquarters)の情報保証部門である通信電子セキュリティグループCES(Communications-Electronics Security Group) 内に設置された，バイオメトリクスに関する検討を専門的に行うWorkingGroupである。内閣府の資金により運営され，国内外の有識者，および英国政府機関の代表者で構成される。プライバシーや人権保護といった政策的な面から，認証精度の評価といった技術的な面に至るまで，電子政府，電子商取引の環境を構築する上で要件となる項目について検討されている。日本では産業技術大学院大学の瀬戸がメンバーとなっている。
TeleTrusT WG6 (Working Group 6) (ドイツ)	TeleTrusT は，情報通信技術全般の信頼性向上を目的に，1989 年に設立された非営利組織 NPO(Nonprofit Organization)である。メンバーとして，情報通信技術に関わる産業界組織の他，連邦情報技術安全庁，連邦刑事庁といった政府組織や技術評価機関，医療機関等，合わせて98の組織が参加している。 技術分野別に9つのWorking Groupが設けられており，バイオメトリクスに関する検討は，技術的な課題のみならず法的な課題も含めた検討，情報共有を行う場として，1997年に設立されたWG6で行われている。これまでの主な活動は，WG6における検討成果を文書化したKriterienkatalog（Critera Catalog）の発行，1999年から2002年にかけて実施した金融機関ATMでの本人確認手段としての，バイオメトリクス利用を想定した技術的，法的問題，インターフェース，社会的受容性の検証を行なう実証実験プロジェクトBioTrusTの実施がある。BIOVISIONでは中心的組織としての役割を果たした。
フラウンホーファー協会 (Fraunhofer Gesellsceft) (ドイツ)	フラウンホーファー協会は，1949 年に設立された非営利組織（NPO）で，ドイツ国内の約60の研究所が所属する他，ヨーロッパ各国，米国，日本，中国，シンガポール，インドネシアにオフィスを構える一大研究グループである。対象分野は幅広く，実用化指向が強い研究組織として知られている。バイオメトリクスに関する取り組みを行なっているのは通信セキュリティ研究所SIT(Fraunhofer-Institut für Sichere Telekooperation)とコンピュータグラフィックス研究所IGD (Fraunhofer-Institut für Graphische Datenverarbeitung)である。 通信セキュリティ研究所では，主に標準化に関する活動が行なわれている。SC37におけるバイオメトリクスに関する国際標準化作業（CBEFF，BioAPI，ISO/IEC 19794）に参加している，この他ドイツ規格協会DIN（Deutsche Institute fur Normung）における国内標準化作業への参加，および薬局で使われるバイオメトリクス組み込みカード“Health Professional Card”の仕様策定を実施している。 コンピュータグラフィックス研究所では，精度評価プロジェクトが実施されている。指紋，虹彩，動的署名の各モダリティについて，詐称への耐性評価を実施している。またドイツ規格協会のアドホックグループ，ISO/IEC JTC1 SC37メンバーとして国内外の標準化活動に参加している。

2） シェンゲン協定とシェンゲン情報システム SIS(Schengen Information System)

シェンゲン協定(Schengen agreement)とは，EUの目的のひとつである物，人，サービス，資本の自由移動が確保された国境なき領域の実現のうち，人の自由移動に関する取り決めである。この取り決めによって，従来の国境管理がすべて撤廃され，人が国境を自由に越えられるようになった。域内を飛行機で移動する際，搭乗者は外国人であっても国内線ターミナルから出発し，税関や入国審査の必要はない。

シェンゲン協定は，1985年6月にルクセンブルグのシェンゲンにて，ベルギー，フランス，ドイツ，ルクセンブルグ，オランダの5ヶ国が調印して始まった。2008年現在，オーストリア，デンマーク，フィンランド，イタリア，ギリシャ，ポルトガル，スペイン，スウェーデン，ノルウェー，アイスランドなど計28ヶ国が協定に調印している。ただし，ノルウェー，アイスランドはEUに加盟していない。

28ヶ国のうち17ヶ国が後述するシェンゲン情報システム SIS(Schengen Information System)を実運用している。EU加盟国では，アイルランドと英国を除いた全てのEU加盟国がシェンゲン協定に調印している。2004年5月にEU加盟を果たしたチェコ，ポーランド，エストニア，ハンガリー，ラトビア，リトアニア，マルタ，キプロス，スロバキア・，スロベニアの10ヶ国もシェンゲン協定に調印している。

しかしシェンゲン情報システムを利用した国境管理の枠組みに参加するには，他加盟国との法規の相互認証や信頼関係構築などが必要である事，またシェンゲン情報システム SIS(Schengen Information System)自体のシステム性能に技術的制限があるため上記10カ国は枠組みに参加していない[12]。

このシェンゲン協定に基づいた，人の自由移動を実現しているのが，シェンゲン情報システム SIS(Schengen Information System)である。各国が提供する個人や財産に関するデータベースを接続し相互閲覧を可能にすることで，域内国境の撤廃に伴い強化された域外国境での出入国管理と，域内での治安維持を支えている。

各国が提供する個人や財産に関するデータベースを接続し相互閲覧を可能にすることで，域内国境の撤廃に伴い強化された域外国境での出入国管理と，域内での治安維持を支えている。

シェンゲン情報システムは，協定を施行している15カ国にアイルランドと英国を加えた17カ国で運用されている。シェンゲン情報システムは，各加盟国別の国内データベースに登録された情報が，フランス，ストラスブールにある中央システムを介して，他国システムに反映され全データベースが同期している。このように中央データベースを持たない分散同期型の仕組みにより，形式上国家主権の分離が維持されている。情報の閲覧は，主に国境の警備，検査，その他の警察，税関検査を目的とした機関に限定される。いかなる場合に警告対象として登録するか，またそれを取り消すかを決定するのは加盟国の権限および裁量である。つまりシステムは協定加盟国共有のものでありながら，そこに蓄積される情報は各国内の個別ものであるといえる。シェンゲン情報システムに登録される対象および情報は表6-11，表6-12の通りだが，旧式のシステムであるため文字情報が主体である。顔写真や指紋などの詳細な情報が必要な場合は，補助機関である SIRENE(Supplementary Information Request at the National Entry)オフィスが個別に対応している[13]。

第2世代シェンゲン情報システムは，ビザ情報システムVISや，2003年1月から稼動している，加

盟国において拒否された亡命希望者による亡命の再申請，いわゆる亡命申請荒らし対策のため，亡命希望者や不正規な方法によってEU域外国境を越えた者の指紋を採取，登録しているEURODACシステムなどとの将来的な接続も考慮もされている。このことから，市民活動家の間では，第2世代シェンゲン情報システムが将来的に巨大な監視システムとして機能するのではないかとの懸念も指摘されている［14］。

表6-11　シェンゲン情報システムの登録対象者および登録情報

対象者	登録情報
・犯罪人引渡しのために手配中である者 ・シェンゲン領域内への進入を拒否すべき者 ・行方不明者・保護を必要としている者 ・刑事事件または法廷に召喚されている証人など ・監視下に置かれるべき者・重大な刑事事件の容疑者として特別な捜査対象となっている者	・　名とすでに登録されている可能性のある偽名 ・性別・身体的特徴 ・出生地と誕生日・国籍 ・武器を所持しているか，暴力的か，あるいは逃亡したか否か ・警告の理由 ・採るべき行動

注）上記文章中，調印と枠組みへの参加とあるが，以下の意味である。

調印：　国境管理を自国からシェンゲン協定加盟国全体での管理へ切り替えを行うことへの参加表明である。

枠組みへの参加：　SISを利用して実際に国境の共同管理を行うことである。

シェンゲン領域：　EU 加盟国中のオーストリア，ベルギー，デンマーク，フィンランド，フランス，ドイツ，イタリア，ギリシャ，ルクセンブルク，オランダ，ポルトガル，スペイン，スウェーデンに加え，協力協定を結ぶノルウェー，アイスランドにより構成される範囲，つまり，SISを運用して実際に国境の共同管理を行っている範囲をシェンゲン領域と呼ぶ。

（2）　MTIT(Minutiae template interoperability testing)プロジェクト

MTITプロジェクトは，欧州委員会 EC（European Commission）の第6期研究開発プログラム(FP6)において設立された14ヶ月間のプロジェクトである。企業，研究機関などから指紋認証研究の第一人者を集め各種検討が行われた。2005年1月から活動し現在は終了している。

プロジェクトの目的は，指紋認証に関わるもので，特に相互運用性（Interoperability）を中心としている。具体的な活動内容は次にあげる6つのワークパッケージとしてまとめられている。

- WP1　組織間マネジメント活動
- WP2　指紋データベースの構築
- WP3　相互運用性試験システムに関する基準の定義
- WP4　指紋認証の相互運用性に関するテストベッドの開発
- WP5　相互運用性試験システムの改良
- WP6　次世代相互運用性試験システムの準備

ヨーロピアンバイオメトリックフォーラムEBF，イギリスの国立物理学研究所NPL（National Physical Laboratory），ドイツのフラウンホーファー協会　通信セキュリティ研究所（Fraunhofer SIT）などの機関に加え，NEC，Motrola，Daon，cogent systemsなど多数の企業がパートナーとして参加している[15]。

（3）BioTesting Europeプロジェクト

BioTesting Europeプロジェクトは，欧州委員会EC（European Commission）の第6期研究開発プログラム(FP6)において設立された9ヶ月間のプロジェクトである。Max SnijderがCEOを務めている。このプロジェクトの目的は，EU域内の国家間で相互運用可能な，パスポートやビザ，IDカードなどの大規模本人確認システムの構築のために必要な，生体認証に関連する技術，システム試験，認証制度の整備にある。Max Snijderの講演資料内の表現では，生体認証に関するパフォーマンス試験，相互運用性試験，セキュリティ評価を行うEU各国の研究所のためのフレームワークを準備することとされている[16]。

プロジェクトは，2007年1月1日より活動を開始しており，活動期間は12月までである。 EBFが運営を指導，その他にイギリスのNPL，ドイツのFraunhofer IGDなどが運営サポートとして組織されている[17]。

2007年10月発表の資料によると，試験機関の組織体制についてはまだ考察段階であるが，EU域内で完結する体制に留まらず，国際的な体制の構築も検討されているという報告がある[18]。

（4）　指紋認証コンテスト FVC(Fingerprint Verification Competition)

指紋認証コンテスト FVC(Fingerprint Verification Competition)とは，イタリアのボローニャ大学バイオメトリックシステム研究所が，米国ミシガン州立大学パターン認識画像処理研究所，サンノゼ州立大学ナショナルバイオメトリックテストセンターと共同で国際的に始めて開催したコンテストである。2007年現在まで2000年，2002年，2004年および2006年と4回開催している[1]。

1）　FVC2000

FVC2000で使用したデータベースは，実環境で採取された指紋サンプルを使用するのでなく，予め定められた方法で採取した指紋画像を使用する。また，参加団体のアルゴリズムのみが試されることから，実システムでの評価とは異なる。データベースの指紋画像は4つの異なるセンサーから採取されたものを使用する。データベース1，2 は小さいサイズの低価格センサー（光学および静電容量型)から採取したものである。データベース3は大型で品質も高い光学センサーから採取したものである。データベース4は人工的に合成された指紋画像を使用している。

各々のデータベースは800種類のテスト画像セットと80種類の画像のトレーニング画像セットに分割され，トレーニング画像は参加団体にアルゴリズムのチューニング用に提供される。

当初25団体からの参加申込があったが，多くの団体が取り下げを表明し最終的に11団体が参加した。取り下げた理由は，評価結果の公開方法などに問題があったと思われる。

表6-12から表6-17に各年度ごとのデータベースの仕様をまとめる。セットA，セットB とある

のは，それぞれテストデータ，トレーニングデータを表し，セットAは100指で各8枚ずつ，セットBは10指で各8枚ずつの画像を有していることを示している。セットBは参加団体にアルゴリズムチューニングのために提供され，主催者側が参加団体から送られたプログラムにセットAのデータを処理することにより性能評価実験が行われる。

表6-12　FVC2000で使用された各データベースの仕様

データベース	センサータイプ	画像サイズ	セットA	セットB	分解能
DB1	低価格光学式センサー Keytronic 社製 Secure Desktop Scanner	300x300	100x8	10x8	500 dpi
DB2	低価格静電容量型センサー ST Microelectronics 社製 Touch Chip	256x364	100x8	10x8	500 dpi
DB3	光学式センサー Identicator Technology 社製 DF-90	448x478	100x8	10x8	500 dpi
DB4	計算機合成指紋	240x320	100x8	10x8	約500 dpi

2）FVC2002

FVC2002の目的およびデータベースの構造等はFVC2000と基本的に大きく変わるところはないが，参加者が企業および研究団体に限定され，匿名による参加が認められるなど，いくつか条件が変更された。最終的に31の団体が参加した。企業が21，研究機関が6，その他が4である。

表6-13　FVC2002で使用された各データベースの仕様

データベース	センサータイプ	画像サイズ	セットA	セットB	分解能
DB1	光学式センサー Identix社製Touch View II	388x374 (142キロピクセル)	100x8	10x8	500 dpi
DB2	光学式センサー Biometrika社製FX2000	296x560 (162キロピクセル)	100x8	10x8	569 dpi
DB3	静電容量型センサー Precise Biometrics社製 100 SC	300x300 (88キロピクセル)	100x8	10x8	500 dpi
DB4	計算機合成指紋 SFinGe v2.51	288x384 (108キロピクセル)	100x8	10x8	約500 dpi

3） FVC2004

結果は2004年2月15日にWEBで公開された。前回までのプログラムとの違いは，照合成績を知らされた後で，匿名か実名を使用するかを選択できる点である。

表6-14　FVC2004で使用された各データベースの仕様

データベース	センサータイプ	画像サイズ	セットA	セットB	分解能
DB1	光学式センサー Cross Match社製　V300	640x480	100x8	10x8	500 dpi
DB2	光学式センサー Digital Persona社製 U.are.U 4000	328x364	100x8	10x8	500 dpi
DB3	感熱式スウィープセンサー Atmel社製 Finger Chip FCD4B14CB	300x480	100x8	10x8	512 dpi
DB4	計算機合成指紋 SFinGe	288x384	100x8	10x8	約500 dpi

4） FVC2006

国立マドリッド・アウトノマ大学(Universidad Autonoma de Madrid)，バイオメトリクスリサーチラボATBS(Biometrics Research Lab－ATVS) が開催グループに加わっている[19]。

表6-15　FVC2006で使用された各データベースの仕様

データベース	センサータイプ	画像サイズ	セットA	セットB	分解能
DB1	Electric Field sensor	96x96 (9キロピクセル)	140x12	10x12	250 dpi
DB2	Optical Sensor	400x560 (224キロピクセル)	140x12	10x12	569 dpi
DB3	感熱式スウィープセンサー	400x500 (200キロピクセル)	140x12	10x12	500 dpi
DB4	計算機合成指紋 SfinGe Ver3.0	288x384 (108キロピクセル)	140x12	10x12	約500 dpi

６．３．４　米国

（1）　背景

1）　組織および活動

米国のバイオメトリクス関連政策は, 2001年9月11日の米国同時多発テロによって大きく転換している。

米国同時多発テロ以前, アメリカにおけるバイオメトリクス研究は, 民間の研究機関が中心で, 政府はそれらの研究に資金援助を行うという形を採っていた。サンノゼ州立大学のNBTC(National Biometric Test Center)や, ミシガン州立大学のパターン認識画像処理研究所 PRIP(Pattern Recognition and Image Processing Lab)が, 米国国防総省 DoD(Department of Defense)からの資金援助を受け, バイオメトリクス技術の精度評価手法Best　Practiceの研究を行っていたことは代表例である[20]。

標準化は, 国立標準技術研究所NIST（National Institute of Standards and Technology）が主導で進めた。また, バイオメトリクス技術の市場への応用を目的とした環境整備を行った。

バイオメトリクスを銀行のシステムに応用する際の安全な管理と運用のためのセキュリティ要件をまとめたANSI X9。84の制定や, 米国内の主要空港で, 商用などで頻繁に入出国する旅客者の利便性向上のため, 入国ゲートに設置されたキオスク端末にて磁気カードと掌形認証による本人確認で入国審査を行う, 入国審査自動化プログラムINSPASSの実施などが代表的である。

米国同時多発テロ以降は, 政府の強いイニシアチブのもとでバイオメトリクス関連政策は, ホームランドセキュリティ分野に集中し, 法律制定に基づく, 体制整備と関連プロジェクトが進められた。2001年10月26日に成立した, テロリズムの阻止と回避のために必要な適切な手段を提供することによりアメリカを統合し強化する2001年の法（Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism Act of 2001), 通称米国パトリオット法(USA PATRIOT Act)を契機に, 政府主導の精度評価プロジェクトの中心組織が, 米国国防総省DoDから商務省直轄の国立標準技術研究所NISTへと移行している[1]。

以下, 米国における活動を大きく国立標準技術研究所NISTおよびINCITS/M1の活動, ホームランドセキュリティ, 米国国防総省DoD(United States Department of Defense)の3つに分類する。

2）　国立標準技術研究所NISTとINCITS/M1

国立標準技術研究所NISTは, 米国商務省DOC（Department of Commerce）の下部組織で, 米国の産業に必須な技術基盤の構築を目的に, 調査研究, 標準化を進める組織である。図3.21に示すように非常に幅広い分野の研究機関を抱えている。バイオメトリクス分野に関わるのは, 主に情報技術研究所ITLと電子電気工学研究所EEELである。その他に製造業者への技術援助など産業界に対する複数のプログラムを抱えている。

また世界最大規模の産官学コンソーシアムである, バイオメトリックコンソーシアムでの活動や, バイオメトリクスに関する標準的なAPIを検討する, BioAPIの活動において中心的な役割を果たすなど, 米国ならびに世界のバイオメトリクスセキュリティに, 最も影響力のある組織の一つである。

米国パトリオット法(USA PATRIOT Act)により, バイオメトリクス分野の体制整備の主要機関

となった後は，2001年11月に米国規格協会 ANSI（American National Standards Institute）認定の標準化組織であるINCITS（International Committee for Information Technology Standards）内にバイオメトリクス専門の技術委員会M1を設置し，標準的な精度評価フレームワークに関する検討を一本化した。

また，ISO/IEC JTC1 SC37の設立に強く関与すると同時に（幹事国は米国），SC37の議長をM1の議長でもあるNISTのFernando Podio　にすることで，国際標準化の主導権を握っている。

NIST内の研究機関を図6-28，バイオメトリクス分野に関連する機関と主な活動内容を表6-16にそれぞれまとめる。

表6-16　NIST内のバイオメトリクス分野に関連する機関

組　織	内　容
情報技術研究所 ITL（Information Technology Laboratory）	NIST内部の研究機関のひとつ。IT技術に関わる精度評価，標準化，新技術の研究を行う
情報システム集約部門 CISD (Convergent Information Systems Division)	情報技術研究所ITLの一部門で，主にデジタルコンテンツの標準を検討していた，バイオメトリクス分野においてはCBEFF，BioAPI標準技術の検討を行った。（2008年現在のNISTの組織体制に同部門は確認できていない）
情報アクセス部門 IAD (Information Access Division)	情報技術研究所ITLの一部門で，他の政府組織やバイオメトリクス製品ベンダーと協力して，指紋認証や顔認証の精度評価プロジェクト，顔画像認証ベンダーコンテストFRVT，指紋認証技術評価FpVTEなどを実施
電子電気工学研究所 EEEL（Electronics and Electrical Engineering Laboratory）	NIST内部の研究機関のひとつ。米国内の電気製品の標準の作成，また標準の測定に必要なテストツールの開発を研究する機関
法執行基準事務所 OLES（Office of Law Enforcement Standards）	電子電気工学研究所EEELの一部門で， Detection, Inspection, and Enforcement Technologiesプロジェクトにおいて，顔画像認証や指紋認証などのバイオメトリクス技術の精度評価の検討を行っている。

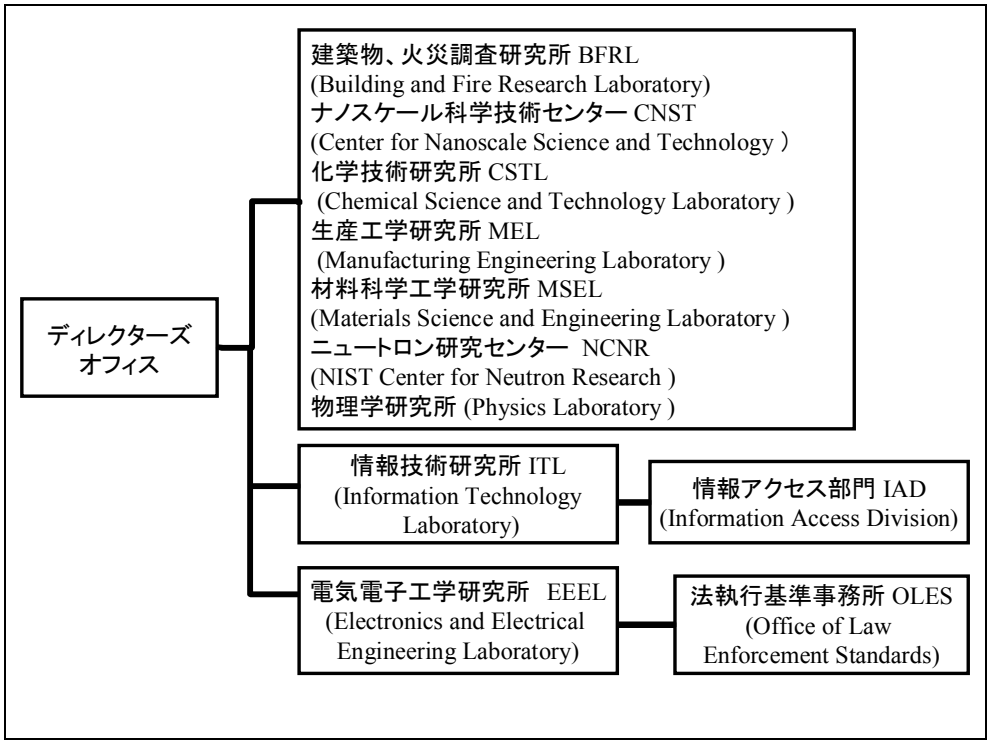


図6-28　NIST内の研究機関

INCITS/M1(International Committee for Information Technology Standards)は，バイオメトリクス技術に関する標準化を専門とする技術委員会で，2001年11月にNISTの要請によりANSI認定の標準化組織INCITS内に設置された。メンバーとして，バイオメトリクスベンダー，インテグレーターの他，国防総省や国務省，国土安全保障省をはじめとする政府組織，大学が参加している。米国におけるバイオメトリクス標準化活動はM1に一本化されており，これまでにBioAPIやCBEFFを国内標準化した実績がある。

この他，現在ISO/IEC JTC1 SC37で検討されている標準化案の新作業項目の多くがM1から提案されており，米国内のみならず国際的な影響力が非常に強い標準化組織である[1]。

2008年1月時点で5つのタスクグループと2つのアドホックグループが設置されている。各タスクグループの活動内容を表6-17に示す。

表6-17　INCITS M1タスクグループの活動内容

タスクグループ	チェアマン	検討内容
M1.2 Task Group on Biometric Technical Interfaces	Fred Herr （ID Technology Partners）	バイオメトリクスコンポーネントとサブシステム間のインターフェイス標準化および安全なデータ転送などのバイオメトリクスセキュリティを担当。さまざまなバイオメトリクス製品との相互運用性を認めるための設計と操作の参照モデルの必要性についての検討
M1.3	Mr. Greg Cannon	バイオメトリクスデータのインターチェンジフォー

Task Group on Biometric Data Interchange Formats	(Cross Match Technologies)	**マット担当　現在活動中のプロジェクト** ・データ互換性のための指紋パターンフォーマット (Finger Pattern Based Interchange　Format) ・データ互換性のための指紋特徴点用フォーマット (Finger Minutiae Format for Data　Interchange) ・データ互換性のための顔認識フォーマット(Face Recognition Format for Data　Interchange) ・データ互換性のための虹彩フォーマット(Iris Interchange Format) ・データ互換性のための指紋画像用フォーマット (Finger Image Based Interchange Fomat, ・データ互換性のための署名画像用フォーマット (Signature/Sign Image Based Interchange Format) ・データ互換性のための掌形用フォーマット(Hand Geometry Interchange Format)
M1。3 Ad-Hoc Group on Data Quality	Mr. David Benini(Aware Inc)	バイオメトリクスサンプルの品質の意味, 表現方法, 品質の解釈について検討
M1.4 Task Group on Biometric Profiles	Mr. Fernando Podio (NIST)	**バイオメトリクスの応用とアプリケーションプロファイルの標準化担当。　現在活動中のプロジェクト** ・アプリケーションプロファイルの相互運用性とデータ互換性　(Application Profile for Interoperability and Data Interchange) ・バイオメトリクスを利用した運輸事業者の個人認証 (Biometric Based Verification and Ientification of Transportation Workers) ・ 相互運用性，データ互換性およびデータ完全性のためのアプリケーションプロファイル (Application Profile for Interoperability, Data Interchange and Data Integrity) ・バイオメトリクスを利用した国境管理のための個人認証 (Biometric Based Personal Identification for Border Management,) ・バイオメトリック認証/識別を用いたPOSシステムのアプリケーションプロファイル(Application Profile for Point-of-Sale Biometric Verification/Identification)
M1.4 Ad Hoc Group on Biometrics and E-Authentication	Mrs. Cathy Tilton,(Daon Inc)	e-authenticationで使用するにふさわしいバイオメトリクスの設計，セキュリティ用件，使用勧告に関するテクニカルレポートの作成を行う責務がある
M1.5 Task Group on Biometric Performance Testing and Reporting	Mr. John Neumann (US Dept. of Homeland Security	**バイオメトリクス精度評価におけるテストとレポートの多面的なテストや報告に関する標準化を担当. M1-5, JTC 1 SC 37WGに対してUS作業グループとして活動**
M1.6	Ms. Elaine M.	**バイオメトリクスの法律及び社会的側面担当**

Task Group on Cross Jurisdictional And Societal Issues	Newton (NIST)	**バイオメトリクスの器具の技術や社会的側面の標準化に関する研究，具体的には，技術，応用，システムの使用の制限，技術以外の課題研究を担当**

出典http://m1.incits.org/

3） 国土安全保障（Homeland security）

国土安全保障省DHS（Department of Homeland Security）は，2002年11月25日に成立した国土安全保障法により，既存の22の政府組織を統合して誕生した，米国の国土保全を目的とした国防総省に次ぐ大規模な省庁である。2001年11月9日に成立した航空運輸保安法により，米国運輸省DOT（Department of Transportation）の下部組織として設立された米国運輸保安局TSA（Transport Security Agency）は，米国国土安全保障省DHSの設立に伴いその下部組織として統合されている[1]。

国土安全保障省DHS，米国運輸保安局TSAは，出入国管理の強化を目的に，外国人の入国時にバイオメトリクスデータの提示を義務付けるUS - VISIT（United States Visitor and Immigrant Status Indicator Technology）プログラムと，共通ICアクセスカードによるバイオメトリクス認証で，運輸機関の制限区域への立ち入りやデータ／システムへのアクセス管理を強化するTWICプログラムの準備を進めてきた]。

2002年5月14日に成立した国境警備強化・ビザ入国改正法に基づき，国土安全保障省DHSは2003年4月9日に，米国に入国する外国人に対して，バイオメトリクス情報の提示を義務付けるUS-VISITプログラムを発表した。US-VISITプログラムでは，入出国時に収集したバイオメトリクス情報を，法執行機関のデータベースと照合することにより，ウォッチリスト（危険人物リスト）との照合，出入国時の同一性確認（入れ替わり防止），滞在期間のチェックの3項目を実施し，セキュリティの強化を図る。US-VISITを構成するアプリケーションを表6-18にまとめる[21]。

表6-18　US VISITに使用されるアプリケーション

アプリケーション名	機　能
ADIS (Arrival Departure Information System)	旅行者の出入国情報を管理
APIS (Advance Passenger Information System)	航空会社と船会社から集められた出入国乗客名簿情報を管理
Claims 3　(Computer Linked Application Information Management System 3)	公的給付金を申請した外国人に関する情報を管理
IBIS (Interagency Border Inspection System)	国際刑事警察機構と全米犯罪情報センターのデータベースをつなぐインタフェース
Ident　(Automated Biometric Identification System)	外国人入国者のバイオメトリックデータを保存
Sevis　(Student Exchange Visitor Information System)	米国に留学している外国人学生に関する情報を格納
CCD (Consular Consolidated Database)	ビザの有効期限やビザの申請履歴などに関する情報を保存

US-VISITプログラムは，2004年1月5日に導入が開始された。2007年12月時点で，116の空港と15の港湾で稼動している[22]。

米国を訪問する外国人は，入国前に空港，港湾で顔画像認証用デジタルカメラと指紋認証用スキャナーを利用し，訪問者のデジタル顔写真撮影と左右人差し指の2指のデジタル指紋を採取される。ビザ免除プログラムVWP (Visa Waiver Program)利用者は，免除されていたが，2004年9月30日以降は，ICAO(International Civil Aviation Organization)認定の，バイオメトリクス情報が含まれている機械読取式旅券MRP (Machine Readable Passport)を保有していない限り審査対象となる。

出国時には，各空港と港湾の出国登録キオスクもしくは，WSA(Work Station Attendant)係員のハンドスキャナーで，米国出国者のデジタル顔写真撮影とデジタル指紋採取を行う。キヨスクもしくはハンドスキャナーにて，渡航文書を読み取り滞在期間等を記録する。出国時US VISIT手続き終了のレシートが発行され，このレシートを乗客ゲートで持参することが必要とされる[23]。

2007年5月6日より，米国を出国する外国人に義務付けられていた出国登録キオスクの使用廃止が発表された。ただし入国時に出入国管理カードI-94フォームをもらった外国人は，現行どおり米国を出国する際に空港，港湾の係員にそのフォームを返さなくてはならない。

国土安全保障省DHSは，これまでの出国手続に代わり，航空業界と連動して米国を出国する外国人への影響を最小限に抑えることができ，かつ出国地点がどこであろうと，外国人の生体認証情報を滞りなく収集できる空港出国時戦略の実施計画案を準備している[24]。

US-VISITプログラムは，以下に該当する人々は審査対象外となっている[25]。

- 14歳未満の子供と79歳以上の人々
- A-1, A-2, C-3, G-1, G-2, G-3, G-4, NATO-1, NATO-2, NATO-3, NATO-4, NATO-5, NATO-6の各ビザ保持者
- 米国国務長官および国土安全保障省DHSに免除された者
- 台湾のE-1ビザ保持者
- カナダ，メキシコ国民の一部

2007年11月29日よりワシントンダレス国際空港で，これまでの2指の指紋採取から10指の指紋採取への移行が試験的に行われている。2008年始めには，新たに表6-19の空港で同様の手続きが開始され，2008年末までにはすべての入国審査所(116の空港，15の海港，154の陸路国境管理ポイント)で実施する予定である[26]。

注）2008年2月1日の日経新聞によれば，2008年1月より正式に10指指紋採取を開始した。

表6-19　2008年に10指の指紋採取が予定されている空港

空港名		所在地
ワシントンダレス国際空港	Washington Dulles International Airport	ヴァージニア州 ワシントン

ボストンローガン空港	Boston Logan International Airport	マサチューセッツ州 ボストン
シカゴオヘア空港	Chicago O'Hare International Airport	イリノイ州 シカゴ
デトロイドメトロポリタン 国際空港	Detroit Metropolitan Wayne County Airport	ミシガン州 デトロイド
ハーツフィールドジャクソン アトランタ国際空港	Hartsfield-Jackson Atlanta International Airport	ジョージア州 アトランタ
ヒューストン空港	George Bush Houston Intercontinental Airport	テキサス州 ヒューストン
マイアミ国際空港	Miami International Airport	フロリダ州 マイアミ
ジョン・F・ケネディ国際空港	John F. Kennedy International Airport	ニューヨーク州 ニューヨーク
オーランド国際空港	Orlando International Airport	フロリダ州 オーランド
サンフランシスコ国際空港	San Francisco International Airport	カリフォルニア州 サンフランシスコ

運輸従事者身分証明プログラムTWIC（Transportation Worker Identification Credential）は，2002年11月25日に発効した海事保安法MTSA（Maritime Transportation Security Act of 2002）に基づき，米国運輸保安局TSAが開発を行ってきた運輸従事者向けの共通アクセスカードである。制限区域や機密データへのセキュアなアクセスの実現と共に，利便性向上による経済効果，プライバシー保護を目的として開発された。カードはJava Cardオープンプラットフォームで FIPS201に準拠している[1]。

TWICプログラムの対象となる運輸従事者は，107ドルから159ドルを支払い顔写真と指紋を記録した身分証明カード（biometrics identification card）を取得しなければならない。また犯罪歴調査，テロリスト警戒リスト（テロリスト監視リスト）との照合や，移民ステータスの確認も実施される[27]。

TWICプログラムは，4期に分けて段階的に実施されている。第1期は，2007年10月16日のウィルミントン港から始まり，12月末までに42の港湾で開始されている。2008年1月から3月までを第2期，4月から6月までを第3期，7月から9月までを第4期として米国全ての港湾で同プログラムが開始される予定である[28]。

US-VISITについてはシステムの改修を行っている最中であり明示的な記述を見つけることはできなかったが，TWICプログラムに使用される生体認証身分証明カードは，FIPS201への準拠がうたわれている事から，６.3.4（2）節で記述したNPIVPまたはFIPS 201 Evaluation Programによるコンフォーマンスを受けていると考えられる。

４） 国防総省DoD(United States Department of Defense)

バイオメトリクスマネージメントオフィスBMO(Biometrics Management Office)は，国防総省内にバイオメトリクス技術を導入する目的で，米国連邦議会の指示により2000年に設立された組織

で，バイオメトリクス技術の開発，選定，および制度化を主導する役割が与えられている。BMOは，内部にバイオメトリクスフュージョンセンターBFC（Biometrics Fusion Center）と呼ばれる研究機関を持ち，バイオメトリクスに関する標準技術や評価基準の開発，および評価の実施に取り組んでいた。

2006年4月に組織が改編され，BMOはBFCと統合してバイオメトリクスタスクフォースBTC(Biometrics Task Force)となっている[29]。Biometric Consortium Confersnceなどの状況から判断すると米国DoDのバイオメトリクスに対する関心が弱まったと推測される。BMOとBFCの事実上の統廃合は，これを裏付けている。

BFCが構築していた製品の評価情報などの活動成果をまとめたデータベースDBKS(DoD Biometrics Knowledgebase System)は，現在DBEKS(DoD Biometric Expert Knowledgebase System)として稼動している。情報の閲覧は登録制である[30]。

旧BMOは，共通アクセスカードCAC(Common Access Card)へのバイオメトリクス導入検討を主導していた。CACとは，国防総省DoD内の施設やネットワークへの物理的／論理的アクセスに対するセキュリティ強化を目的として開発された，GSC-IS準拠の共通アクセスICカードである。2001年11月から米軍関係者全員を対象に配布が開始され，現在までに1300万枚以上が発行されている[31]。

旧BMOやBTCがどのように関与したかは不明であるが，2006年10月からCACの有効期限切れによる更新を利用した次世代CACへの乗り換えが始まっている。次世代CACは，3.4.2節で詳述する国土安全保障に関する大統領令Homeland Security Presidential Directive 12に基づき，それまでのCACでは採用していなかった2指による指紋認証などを採用している[32]。

次世代CACはFIPS201への準拠が求められていることからTWICプログラムで使用されている生体認証身分証明カードと同様に，3.4.2節で記述するNPIVPまたはFIPS 201 Evaluation Programによるコンフォーマンスを受けていると考えられる。

旧BMOは，ISO/IEC15408 コモンクライテリアVer.2.1に準拠したバイオメトリックプロテクションプロファイルの策定を行っている．照合Verificationと識別Identificationのデバイスに分け，さらに使用環境で要求される３段階のロバスト性（Basic，Medium，High）に分けて，5種類のプロテクションプロファイルの策定を進めていたが，このうちのMediumロバスト環境における照合Verificationを対象としたプロテクションプロファイルが，政府組織共通のプロテクションプロファイルとして認定され「US Government Biometric Verification Mode Protection Profile for Medium Robustness Environments」として2003年11月に発行されている[33]。

（２） 連邦政府職員，契約事業者の個人識別情報検証FIPS 201

連邦情報処理標準FIPS (Federal Information Processing Standards)は，軍事以外の全ての政府機関，請負業者による利用を目的として連邦政府が開発した公式の標準規格である。FIPS規格は，連邦情報セキュリティ管理法FISMA(Federal Information Security Management Act of 2002)の成立に伴い，国家安全保障に関係するシステムを除く，すべての連邦政府システムにおいて該当する規格の順守を求める完全強制規定である。

FIPS 201 (Federal Information Processing Standards Publication 201, Personal Identity Verification of

Federal Employees and Contractors)は，2004年8月27日発表の国土安全保障に関する大統領令，連邦政府職員と契約業者の共通識別基準のためのポリシーHSPD-12（Homeland Security Presidential Directive 12）に基づき，国立標準技術研究所NISTのコンピュータセキュリティ部門(Computer Security Division)により作成され，2005年2月25日に公表された個人識別情報PIV(Personal Identity Verification)に関する連邦政府基準である。技術標準の最終承認は商務省長官が行った。改訂版のFIPS 201-1が，2006年3月に発行されている。

各省庁は，HSPD-12によってこの連邦政府の個人識別情報管理標準であるFIPS 201に準拠することが義務付けられ，政府全体において，施設やネットワークへのアクセスに対する一貫したID管理が行われる。

FIPS201は，2部構成となっている。第1部では，ID発行に関する一連のプロセス（人物特定，セキュリティ，プライバシーに関する共通の必要条件）について，第2部では各省庁が発行するカードにいかに技術的な互換性を持たせるかという点，カードの機能設計，システムインターフェイス，データ要素など，例えばカードに接触型/非接触型のチップや，人差し指の指紋情報，PKIのための証明書や暗号アルゴリズムなどを盛り込むことなどについて述べられている。

FIPS 201は，PIVシステムとしての最小要件と詳細要件を定めている。NISTは，FIPS-201 に基づき，米国の政府機関がセキュリティ対策を実施する際に利用することを前提としてまとめたコンピュータセキュリティ関係のレポートSP(Special Publication) 800シリーズに詳細な仕様を作成しており，これがPIVの仕様書類となっている。

図6-29にPIVの仕様に関わる文書の関連図，表6-20に各文書の詳細を示す[34]。

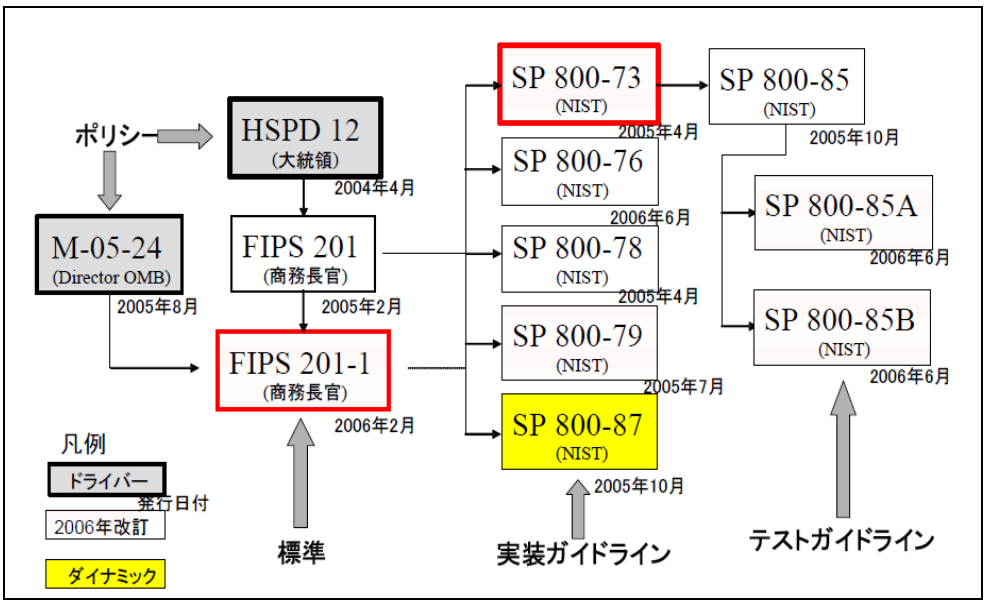


図6-29　PIVの仕様に関わる文書の関連図

表6-20　PIV技術文書の一覧

文書種別と番号	タイトル	概要
FIPS 201-1	連邦政府職員，契約業者のための個人識別情報の検証　(Personal Identity Verification of Federal Employees and Contractors)	PIVシステムにおける最小要件に加え，PIVカードの物理カード特性，ストレージメディア，データ要素について規定された文書
SP 800-73-1	PIV用のインターフェース　(Interfaces for Personal Identity Verification)	証明書の保管に関するインターフェースとカードアーキテクチャ
SP 800-76-1	PIV用のバイオメトリックデータ仕様 (Biometric Data Specification for Personal Identity Verification)	バイオメトリクス情報のインターフェースとデータフォーマット
SP 800-78-1	PIV用の暗号アルゴリズムと鍵長 (Cryptographic Algorithms and Key Sizes for Personal Identity Verification)	PIVで利用される暗号アルゴリズムや鍵長，パラメータなどの仕様
SP 800-79	PIVカード発行組織の認証と認定のガイドライン　(Guidelines for the Certification and Accreditation of PIV Card Issuing Organizations)	認定を望むカード発行者が実施すべきいくつかの属性についての記述
SP 800-85A	PIVカードアプリケーションとミドルウェアインターフェースのテストガイドライン　(PIV Card Application and Middleware Interface Test Guidelines (SP 800-73 compliance)	PIVミドルウェアとPIVカードアプリケーションの2つへの準拠検証に使われるべきテスト要件とテストアサーションについて記述
SP 800-85B	PIV データモデルテストガイドライン (PIV Data Model Test Guidelines)	データモデルに関するテスト要件や，テストアサーション，準拠テストについてのガイドライン
SP 800-87	連邦政府組織・関連組織識別用コード (Codes for the Identification of Federal and Federally Assisted Organizations)	FIPS 201で規定されているカード保有者ID（CHUID）に含まれる連邦政府カード証明書番号（FASC-N）を記述するのに必要な組織コードの一覧
SP 800-96	PIV Card / Reader Interoperability Guidelines	PIVとカードリーダ間の相互運用可能性を確保するためのガイドライン
NISTIR	7284 Personal Identity Verification Card Management Report,	SP 800-78-1にはPIVカード管理の仕様は含まれていない。カード管理システムの概要を示すと同時に，今後のPIVの仕様変更，追加を示唆している
NISTIR7337	Personal Identity Verification Demonstration Summary	NISTが企画したPIV カード等，FIPS-201をサポートした製品のデモンストレーションの報告書 HSPD-12/FIPS-201の実装状況を報告している

FIPS-201の基準に関連した製品，サービスは多岐に渡っているが，これらの製品やサービスの評価(Evaluation)と認定(Approval)を行なう必要がある。NISTは，FIPS-201に準拠したPIVカードとミドルウェアを検証して，認定するNPIVP(NIST Personal Identity Verification Program)を立ち上げた。

図6-30に示すように，NPIVPは，PIVカードとミドルウェアがSP800-73の仕様への適合についてテストと検証を提供している。テスト仕様としては，SP 800-85Aが使われテストが行われる。テスト自体は，NPIVPにより公認されたラボによって行なわれる。

尚、SP800-76-1ではISO国際規格のベースになったANSI規格が参照されている。

PIVミドルウェアのテスト，検証，認定の手順は以下の通りである[34]。

１）　ベンダーは，ラボと契約を結んで，テストするためにPIVミドルウェアをラボに提出

２）　ラボではPIVミドルウェアをNPIVPのテスト・ツールキットでテストして，テスト報告書を作成

３）　テストラボは検証のためにNPIVPにテスト報告書を送付

４）　ラボを通じてベンダーへ認定書(certificate)を送付

PIVカードに関しては，NPIVPによる認定だけではなく，米国政府機関が暗号製品を購入する場合の調達基準であるFIPS-140の認定取得が必要になる。FIPS-140の認定は，FIPS -140に基づいた暗号モジュール評価制度であるCMVP(Cryptographic Module Validation Program) により行なわれる[33]。

連邦政府調達局GSA(General Services Administration)でもFIPS 201 Evaluation ProgramというFIPS 201への適合性を評価する制度を実施している。図6-30に示すNPIVPの試験体制と同様に第三者機関であるラボで試験を行う第三者適合性評価体制をとっている。

NPIVPとの違いは, NPIVPがPIVのアプレットとミドルウェアの評価のみを行うのに対して, FIPS 201 Evaluation Programではバイオメトリックリーダー, テンプレートジェネレーター, テンプレートマッチャー，指紋採取ステーション，指紋スキャナー，顔画像採取ステーションなど22のカテゴリに対応している点である[35][36]。

2006年6月にラボが活動を始めてから2008年1月現在まで332の製品が認証され, そのリストが公式サイト上(http://fips201ep.cio.gov/apl.php)に掲載されている。

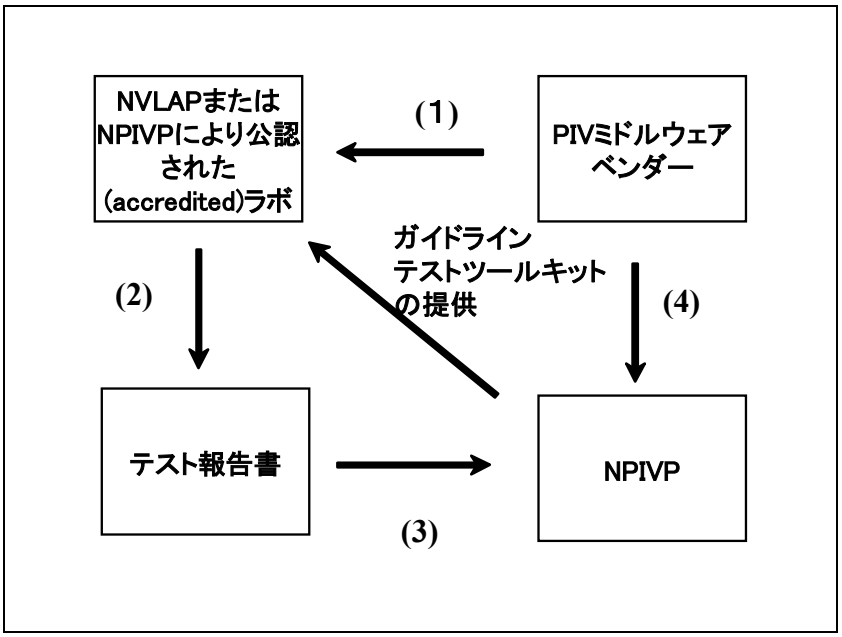


図6-30　NPIVPの認定制度

（3）　コンテスト

米国では，新技術の開発や新技術を実用段階まで引き上げるための手段として政府機関の出資による開発促進プログラムや，企業や研究機関向けの精度評価コンテストが行われている。

精度評価コンテストの主体は，国防総省やNISTといった大口の政府調達にかかわる機関であるため，必然的にコンテストの結果を巡ってベンダーの競争が生じる。この構造は，ISO17000に照らし合わせてみると一種の第2者認証であるということができる。

１）指紋認証コンテスト FVC(Fingerprint Verification Competition)

サンノゼ州立大学ナショナルバイオメトリックテストセンターとミシガン州立大学パターン認識画像処理研究所がイタリアボローニャ大学と共同で国際的に始めて開催したコンテストである。詳細は6.3.2.3欧州の項を参照。

２）顔画像認証ベンダーコンテストFRVT(Face Recognition Vendor Test)

FRVT(Face Recognition Vendor Test)は商用の顔認証技術を対象とし，複数の政府機関が共同出資して行っている精度評価コンテストで，これまでに2000年，2002年，2006年の3回開催されている。以下にFRVT2000，FRVT2002，　FRVT2006の特徴をまとめる[33]。

①FRVT2000

国防総省と司法省が共同出資して開催した，商用の顔認証技術を対象とする初めての精度評価コンテストである。評価用データベースはFERETプログラムを拡充したものが使用されている。データベース中の画像全ての単純マッチングの他，アクセスコントロールシステムを想定したシナリオに基づき，登録および経時変動をシミュレートしたテストが実施された。市販製品の評価

プロジェクトとしては先駆的な試みであった[33]。

以下の5団体が参加している。

- Banque-Tec International Pty Limited
- C-Vis Computer Vision und Automation GmbH
- eTrue (2002年6月ヴィサージュ(Viisage)に吸収)
- Lau Technologies
- Visionics Corporation (2002年7月アイデンティクス(Identix)に吸収)

②FRVT2002

NISTの主導の下で実施された精度評価コンテストである。国防高等研究計画局DARPAや連邦捜査局FBIなど国内外16の組織がスポンサーとして参加した。データベースも被験者数3万人規模，総画像枚数も10万枚規模と飛躍的に大規模となった。原画像だけではなく，3次元変形モデルによって姿勢変動を補正した画像による精度評価を行った点が特徴である[33]。

以下の10団体が参加している。

- AcSys Biometrics Corp.
- Cognitec Systems GmbH
- C-VIS Computer Vision und Automation GmbH
- Dream Mirh Co., Ltd.
- Eyematic Interfaces Inc.
- Iconquest
- Identix
- Imagis Technologies Inc.
- Viisage Technology
- VisionSphere Technologies Inc.

③FRVT2006

FRVT2002に続きNIST主導で行われた。産学界から認証技術を公募して次世代の顔認証技術を対象に高精細画像や3次元形状のデータベースによる精度評価，顔認証技術開発プログラムFRGCの成果検証を行った[37]。

以下の10カ国22団体が参加している。

- Animetrics, Inc.
- Carnegie Mellon University
- Cognitec Systems GmbH
- Diamond Information Systems
- Geometrix, Inc.
- Guardia
- Identix, Inc.
- Neven Vision

- New Jersey Institute of Technology
- Nivis, LLC
- Old Dominion University
- Panvista Limited
- Peking University, Center for Information Science
- PeopleSpot Inc.
- Rafael Armament Development Authority Ltd.
- SAGEM SA
- Samsung Advanced Institute of Technology
- Tsinghua University
- Tili Technology Limited
- Toshiba Corporation
- University of Houston
- Viisage

３）指紋ベンダー技術評価 FpVTE (Fingerprint Vender Technology Evaluation)

FpVTEは，国立標準技術局NISTが主催する指紋認証に関する精度評価コンテストである。評価の方法・分析手法に関しては，FRVT2002を模範としている。2003年5月～11月に実施された。司法省がスポンサーとなっている[33]。

テストの目的は以下の通り。

- 運用を想定したデータを利用した指紋認証システムの精度評価
- 最も高精度な指紋認証システムの選定
- 被験者要因による変動が精度に与える影響の調査
- データベースの再構築（より効率的な評価用データベースの構築）

認証速度，スループット，センサー等の入力デバイス，コストに関しては，評価の対象外とされている。

評価に使用された指紋データベースは，連邦政府機関や州政府機関から集められた25309人，総計39370指の画像からなる。データベースには，以下の３種の指紋画像が含まれている。

- 通常の単一指紋画像（プレーン指紋画像）
- ４本同時に撮られた画像から切り出された指紋画像
- 指を回転させて撮影した指紋画像（ロール指紋画像）

本コンテストには，以下に示す18ベンダーの34システムが参加している。

- 123 ID, Inc.
- Antheus Technology, Inc.
- Av@lon Systems Inc.
- BIO-key International, Inc.
- BioLink Technologies International, Inc.

- Bioscrypt, Inc.
- Cogent Systems, Inc.
- Dermalog Identification Systems GmbH
- Golden Finger Systems
- Griaule Tecnologia
- Identix, Inc.
- SAGEM MORPHO, Inc.
- NEC
- Neurotechnologija Ltd.
- The Phoenix Group, Inc.
- Motorola
- Raytheon Company
- Technoimagia Co., Ltd
- Ultra-Scan Corporation

４）顔認証技術開発プログラムFERETおよびFRGC

米国では政府支援で顔認証技術の開発プログラムが実施され，その一環として評価用の大規模データベースの構築が行われている。本節では，1990 年代に国防総省主導で実施されたFERET プログラム，および，現在NIST 主導で実施されているFRGC について概要を紹介する[33]。

①FERET（Face Recognition Technology）

FERETは，1993年から1997年の間，国防総省の主導の下，650万ドル以上の資金を投じて実施された顔認証技術開発プログラムで，顔認証技術研究への資金助成，認証技術評価の為の大規模データベースの構築，評価の実施を行っていた。

FERETプログラムでは，約1000名の被験者について，照明環境や顔の向きを変動させて撮影した約14000枚のグレースケース顔画像のデータベースを構築し，産学機関から公募した認証技術の評価を行った。FERET データベースは，現在NISTの管理の下，カラー画像を加えた形式で無償公開されている。

②FRGC (Face Recognition Grand Challenge)

FRGCは，FERETに続く次世代の顔認証技術開発を目的としたプログラムで，NIST主導の下で2004年5月から2006年3月まで実施された。

公募の形式で広く研究者を募り，次の3 種の認証技術を開発の対象とする。

- 従来比4～6倍の解像度の高精細画像をベースとした顔認証技術
- 3次元形状データをベースとした顔認証技術
- 照明変動や姿勢変動を補正する前処理技術

認証技術の開発の一環として，次の2項目が実施されている。

- 大規模顔データベースの構築
- 実験環境や結果を記述するXML ベースの共通フレームワーク（BEE）の構築

５）虹彩評価チャレンジICE（Iris Challenge Evaluation）

虹彩評価チャレンジICEは，国立標準技術局NISTと複数の政府系機関の出資による虹彩認証技術の開発促進のために開催したコンテストである[38]。

これまでに2005年と2006年の2度に渡り開催されている。ICE2005は2005年8月から2006年3月まで行われ，虹彩認証技術の開発に重点がおかれていた。2006年6月から行われたICE2006は，企業，大学，研究所などからの虹彩認証アルゴリズムの精度評価試験への参加を募り，虹彩認証技術としては初めての大規模なコンテストを行った。

以下は最終的にICE2006に参加した8団体である。

- Carnegie Mellon University
- Chinese Academy of Sciences Institute of Automation
- SAGEM and Iridian Technologies, Inc.
- IriTech, Inc.
- JIRIS USA
- University of Cambridge
- University of West Virginia
- Tohoku University and Yamatake Corporation

ICE2006の詳細な結果については，国立標準技術局NIST情報アクセス部門IADのサイト内に掲載されているFRVT 2006 and ICE 2006 Large-Scale Resultsにまとめられている[39]。

６）MBGC(Multiple Biometric Grand Challenge)

MBGCは，FRVT，FRGC，ICEの各コンテストの結果を受けて，低解像度のスチル写真の認識や動画イメージの認識，様々な照明環境下での顔，虹彩画像認識，顔の一部が遮蔽されている画像や非正対顔画像の認識などこれまでのコンテストでは検討されていなかったより実運用に近い環境下での問題を検討することを目的として国立標準技術局NISTと複数の政府系機関の出資により設置された[40]。

2007年12月に告知され2008年1月現在もMBGCのサイト上(http://face.nist.gov/mbgc/)で参加者を募っている。2008年の3月から4月にかけて最初のワークショップが開催される予定である。

具体的な検討課題は以下である。

- 現実世界に近い高解像度と低解像度の顔画像の認識
- 動画やオフアングルの画像からの虹彩認識
- 顔画像と虹彩を融合させた認識技術
- 認識のための撮影制約を設けていない顔画像，動画の認識
- 赤外線や高解像度撮影による動画の認識

（4） IBG

IBG(International Biometric Group)は，1996年設立，ニューヨークに本社のあるバイオメトリクスセキュリティに関する米国の民間企業である。コンサルティングやシステム構築，調査レポー

トの作成，試験認証サービスの実施など幅広い活動を行っている。

コンサルティング分野では，政府機関や地方政府，民間企業などに対して，バイオメトリクスを利用した認証システム構築のためのRFPやビジネスケースの作成，プロジェクトマネジメントやソフトウェア独立検証および妥当性確認(IV&V)，PIA(Privacy impact assessments)などを行う。

調査レポート分野では，バイオメトリクス試験認証サービスの比較評価，既存システムのセキュリティ脅威レベル評価，バイオメトリクス市場や技術関連レポート作成などを行う。

試験認証サービスは，ANSI INCITS 409.1-2005(Biometric Performance Testing, Principles and Framework)，ANSI INCITS 409.3-2005(Biometric Performance Testing, Scenario Testing and Reporting)，ISO/IEC 19795-2(Testing Methodologies for Technology and Scenario Evaluation) の各標準に適合したCBT (Comparative Biometric Testing)という手法を使用している。

CBTによりセンサ端末からソフトウェアや照合アルゴリズムの性能評価を包括的に行う。またBiometric Performance Certification Programにより企業のバイオメトリクス製品の各種性能指標を4段階に分けて評価，認定を行っている。図3.24はその一例である。この図からユーザビリティとして本人拒否率，正確性として他人受入率に関する性能を数値化し測定していることが考えられる[41]。

精度評価については国際標準に準拠した第三者コンフォーマンステストを実施している。

Usability		**Certified Level 3**
Failure to Enroll Rate (FTE)	2.53%	Level 3
Average Enrollment Transaction Duration (Seconds)	48.05	Level 3
Transactional Failure to Acquire Rate (T-FTA)	0.32%	Level 4
Average Recognition Attempt Duration (Seconds)	5.05	Level 3
Accuracy		**Certified Level 3**
False Match Rate (FMR)	0.00001%	Level 4
False Non-Match Rate (FNMR)	1.980%	Level 3
Transactional False Match Rate (T-FMR)	0.00035%	Level 4
Transactional False Non-Match Rate (T-FNMR)	1.038%	Level 3

図6-31　Biometric Performance Certification Program結果の例

IBGの具体的な活動実績として，以下の例が紹介されている。

- カリフォルニア州車両局のIDシステム導入のためのRFP作成，および技術支援
- 連邦政府へのマルチモーダル認証システム導入
- 国立司法研究所NIJ(National Institute of Justice)へのマルチモーダルシステムの贈与
- 大統領府科学技術政策局OSTPの依頼によるUS-VISITの長期戦略
- 運輸保安局TSAの依頼によるテクノロジーロードマップ作成
- テキサス州衛生福祉サービス委員会の医療用スマートカードのソフトウエア独立検証および妥当性確認プログラム
- 国土安全保障省DHSの依頼による虹彩試験認証技術調査
- 香港入境管理事務所のPIA調査

６．３．５　日本

（1）　背景

1）　組織および活動

日本では，1996年から1998年にかけて，次世代電子商取引推進協議会(ECOM)に設置された本人認証技術検討ワーキンググループにて，本人認証技術としての評価基準などの検討が行われたのを皮切りに，1998年から1999年にかけてIPAの「指紋照合統一規格API および指紋照合システムの評価基準の策定」で指紋照合システムの評価基準，指紋照合統一規格APIが産業界中心のワーキンググループにより検討された。具体的な参加企業は表6-21の通りである[41]。

表6-21　指紋照合統一規格API および指紋照合システムの評価基準の策定参加企業一覧

指紋照合システムの評価基準	指紋照合統一規格ＡＰＩ
株式会社日立製作所（取り纏め）	システムニーズ株式会社
オムロン株式会社	ソニー株式会社
ソニー株式会社	三菱電機株式会社
三菱電機株式会社	翼システム株式会社
	オムロン株式会社
	株式会社東芝
	シュルンベルジェ株式会社

「指紋照合統一規格APIおよび指紋照合システムの評価基準の策定」の成果を受けて，INSTACバイオメトリクス標準化調査研究委員会にてJIS標準情報が策定されるなどの活動が行なわれてきた。また，ネットワークサービスを念頭に置いたバイオメトリクス認証に注目した活動を行なう業界団体である日本自動認識システム協会内にバイオメトリクス部会が設立されるなど，産業界主導で業界団体が結成され，様々な活動が行なわれてきている。

2002年に設立されたISO/IEC JTC1 SC37に対して，日本では2002年10月にSC37国内委員会，2003年4月にユビキタスネットワーク社会におけるバイオメトリクスセキュリティ研究会(BS研究会)，2003年6月にバイオメトリクスセキュリティコンソーシアムBSC (Biometrics Security Consortium)が設立され，産官学連携による検討体制で対応に望んでいる。表6-22に継続的に運営されている組織の主な活動についてまとめる[42]。

表6-22バイオメトリクス分野に関わる主要な組織

組織	活動　内容
ISO/IEC JTC1/SC37	（社）情報処理学会　情報規格調査会に国際のSC37に対応する国内専門委員会が2002年秋に設置された。６つのWGから構成される（委員長：産業技術大学院大学瀬戸洋一，幹事：ソニー　向井幹男）
バイオメトリクスセキュリティコンソーシアム BSC (Biometrics Security Consortium)	バイオメトリクスセキュリティコンソーシアムBSCは，バイオメトリクス技術や，それに関連する社会制度に対し産業界が抱える課題の洗い出しと，バイオメトリクス技術の導入を促進する市場環境整備のために，2003年6月に設立された団体である。現在組織は，（社）日本自動認識システム協会の一部の組織となっている（議長：産業技術大学院大学瀬戸洋一）。 2007年5月設置されたAsia Biometoric Consortium（2001年9月に日本の提案で設置されたAsia Biometric Forumが発展的解消し設置された組織）の日本の対応機関となっている（議長：中国科学技術院のTien Tan，実行委員：瀬戸洋一）
バイオメトリックシステムセキュリティ研究会	前身である「ユビキタスネットワーク社会におけるバイオメトリクスセキュリティ研究会(BS研究会)」は，2003年4月に電子情報通信学会通信ソサイエティに新設された。 2006年に研究会名をバイオメトリックシステムセキュリティ研究会とした（委員長：産業技術大学院大学　瀬戸洋一）。 活動のスコープは，バイオメトリクスに関するすべてをテーマとする学術的な研究活動を行うことにある。パターン認識とセキュリティ，センサとアルゴリズムなど境界領域での議論が重要となり，これらの領域の専門家が一堂に会して領域横断的な議論をする場が必要である。BS研究会はそのような要求を満たす貴重な研究会に位置づけられる。
(社)日本自動認識システム協会 バイオメトリクス部会	日本自動認識システム協会は，バーコード，2次元シンボル，RFID, バイオメトリクス等の自動認識システムに関する調査研究や標準化の推進，普及および啓発行うことにより，物流，流通等のシステムの効率化，高度化の推進に貢献することを目的として，1999年2月に社団法人化された。 産業分野毎に設置された6部会のうちの1つとしてバイオメトリクス部会がある。バイオメトリクス部会では，バイオメトリクス産業のポータルサイト，市場統計調査，啓発のためのセミナー，プライバシーガイドラインの調査を取組テーマに掲げている（部会長：ICS　宇都宮康夫）。

（2）　外国人生体情報システムBICS

日本では，2006年に出入国管理及び難民認定法が改正され，外国人が入国する際の指紋採取，顔写真撮影などの個人識別情報情報提供が義務化された。これを受けて出入国を管理する法務省入国管理局と警察庁を結ぶ外国人生体情報システムBICSの構築が検討された。

BICS構築の参考事例としてはUS-VISITがとりあげられており，検討項目も予算，ベンダーの選

定，ネットワークなどのインフラ環境まで多岐に渡っているが，重要な課題として警察庁と法務省の間の指紋データの相互運用性があげられている。

採用するシステムはインターフェース規格としてBioAPI，データ規格としてCBEFFへの準拠が予定されているが，警察庁は指紋情報を回転指紋で10指保持しているのに対し，法務省は平面指紋で2指のみ保持しているなど組織ごとの保有情報に差異がある点，システム構築ベンダーが異なることによる指紋照合アルゴリズムの違いに対する互換性の確保の問題，指紋照合における他人受け入れ率と本人拒否率の設定の問題など生体認証に関わる多くの検討課題が挙げられている[43]。

2008年1月現在入手できる情報では，上記システムの検討が有識者により行われた事実のみである。。しかし2007年11月20日から改正出入国管理及び難民認定法の施行により，出入国時の指紋と顔写真撮影が実施されていることから，外国人生体情報システムBICSは既に実稼動していると考えられる[44]。

（3） 船員データ電子化１検討委員会

１） 概要

米国同時多発テロ以降，ホームランドセキュリティの強化が世界的な潮流となるなか，海事分野においても国際労働機関ILO(International Labor Organization)が，第185号条約(船員の身分証明書条約)を2003年6月に採択，バイオメトリクスを用いた船員の身分証明書の導入が盛り込まれた。

この潮流に対し日本では2004年度から国土交通省が，東京学芸大学野川忍教授を座長とし学識経験者や関係官庁による船員データ電子化検討委員会を設け，ILO第185号条約に定められた船員の身分証明書の要件を満たす船員データの電子化システムの構築と次世代船員手帳の検討を行っている。以下に年度ごとの活動内容をまとめる。

(A) 2004年（平成16年度）

船員データの電子化による海事保安の強化および船員行政の効率化を実現するため，2005年（平成17年度）の試行的導入を視野において以下の事項を満たすシステムの設計動作試験を完了する。

- 1000人分の船員の　名，住所，雇入契約の内容等がデータベースに蓄積できること
- 船員手帳に船員の　名，住所，雇入契約の内容等をプリンタで印字できること
- 船員の　名，住所，雇入契約の内容等が船員手帳等のＩＣチップに搭載できること
- 船員手帳等に指紋情報がバーコードとして印刷できること
- バーコード及びICチップの読み取り装置が監査の際に携帯できること

(B) 2005年（平成17年度）

2004年(平成16年度)に設計したシステムについて試行的導入（船員手帳を1000人の船員に試行的に保有，試用してもらう）を行い，平成16年度に設計したとおり実際に機能するかどうか確認する。試行的導入結果を元に，ILOで規定されている数値を上回る以下の目標を達成する。

- 誤認率（他人受入率） 0.001%以下
- 誤認率（本人拒否率） 0.5%以下

- 船員手帳交付手続所要時間10分以内
- バイオメトリクス情報の登録時間45秒以内
- バイオメトリクス認証時間10秒以内

2） システムの基本設計

2004年(平成16年度)においては，システムの基本設計に関し，海事保安強化のための指紋認証機能，船員行政効率化のための船員手帳発給，雇入事務処理機能，船員労務監査のモバイル通信機能，乗船履歴確認機能を備えたシステムの基本設計が完了している。

2005年(平成17年度)においては，前年に構築したシステムの基本設計に基づく実証実験システムを構築し，同システムによる政策目標の達成度合いを検証するための実証実験を行っている。具体的には，平成18年1月20日から3月10日までの間，関東運輸局，東京運輸支局(青海庁舎)，中部運輸局，近畿運輸局，神戸運輸監理部において，1000名を目標とする協力者を対象として実施した。実証実験による船員手帳，身分証明書については，船員法に基づく効力を有しない試作品として発行した。1048人の協力者による実証実験により，政策目標について以下の通り全て目標値を上回る結果が得られた。

- 誤認率（他人受入率） 0.0007% （目標：0.001%以下）
- 誤認率（本人拒否率） 0.1% （目標：0.5%以下）
- 船員手帳交付手続所要時間3.6分 （目標：10分以内）
- バイオメトリクス情報の登録時間28.7秒 （目標：45秒以内）
- バイオメトリクス認証時間6.5秒 （目標：10秒以内）

3） 次世代船員手帳

次世代船員手帳は，活字ベースの情報をＩＣチップに記憶することや，バイオメトリクス情報をバーコード化などの船員データの電子化により，船員手帳交付手続所要時間短縮による行政事務の効率化，他人受入率，本人拒否率，バイオメトリクス情報の登録時間，バイオメトリクス認証時間に関する各種政策目標を達成し，本人確認の確実性の向上を図ることが目的である。

次世代の船員手帳については，その検討において，事務局より手帳型，バインダー型，カード型，手帳・カード併用型及びバインダー・カード併用型の試作案が提示され，それぞれについて検討が行われた。その結果，身分証明機能と労働保護に関する機能をそれぞれ分離し，身分証明機能はカード形式に，労働保護に係る機能は表紙にＩＣチップを埋め込んだバインダー形式とし，両方を一体で保持することができる形式として実証実験が行われた[45]。

６．３．６ まとめ

（1）適合性評価機関が公式に設置されているのは，韓国のみである。韓国は欧米の情報を的確に収集し，先進的な取り組みに活かしている。韓国における評価機関の設置の目的は，自国の市場が小さいため，バイオメトリクス製品を国際市場へ展開するためのひとつの重要な戦略として位置づけている。アジアでは，適合性評価への取り組みは，韓国，シンガポールで熱心である。ただし，適合性評価を実施するに対応可能な産業がないため，政府や大学が先行してい

るに過ぎない。

（2）欧米では，国際標準に先駆けて，自国（あるいは共同体内）で実質的な適合性評価ノウハウを獲得している。具体的には，EUにおいては，SIS-Ⅱシステムの構築に向けたBiotTesting Euorope プロジェクト，米国においては，FIPS 201 (Federal Information Processing Standards Publication 201, Personal Identity Verification of Federal Employees and Contractors)の基準に関連した製品，サービスの評価(Evaluation)と認定(Approval)を行なうNPIVP(NIST Personal Identity Verification Program)およびFIPS201 Evaluation Programを立ち上げている。

つまり，欧米では，国際標準と関係なく，その必要性から，適合性評価のフレームワークをつくり，いつでも国際標準に整合させることのできる，ノウハウと体制を構築している。

（3）日本においては，政府関係の電子パスポート，入国管理システム，民間では金融関係システムで，大規模なシステム構築が行われたが，残念ながら，国としての戦略的な政策にかけていたため，適合性評価組織をつくためのノウハウなどが蓄積されていない。

参考資料

[1] 情報処理推進機構IPA: 各国バイオメトリクスセキュリティ動向の調査 2004.2

[2] 佐々木朋美: マイコミジャーナル-韓国技術標準院，2010年までに205技術の国際標準採択を目指す5カ年計画を発表- 2005.8
http://journal.mycom.co.jp/news/2005/08/16/001.html

[3] 瀬戸洋一編:ユビキタス時代のバイオメトリクスセキュリティ2003.1

[4] Jason Kim: KISA at a glance 日本自動認識システム協会における招聘講演 2007.8

[5] Youngjun Kim: Schemes and Procedures of K-NBTC for conformance Test(CTS Service of K-NBTC), ABC2008 2007.9

[6]CICCシンガポール事務所: 東南アジアIT標準化調査
http://www.cicc.org.sg/ciccrp/C3ir0506.htm 2005.6

[7] 国際通信経済研究所RITE: シンガポールの情報通信基盤整備政策 iN2015
http://www.rite-i.or.jp/kenkyuin/telecom/repo0704.html 2007.4

[8]シンガポール総務省: INAUGURAL ASIAN BIOMETRICS CONSORTIUM CONFERENCE & EXHIBITION https://www.governmentware07.com/ABC07/abc07_conformance.htm

[9] Yau Wei Yun, Chen Tai Pang Lawrence: ISO Standards on General Biometric Technologies & Conformance Testing Institute for Infocomm Research

[10] Esprit 21978 - BIOTEST Biometric Testing Services
http://cordis.europa.eu/esprit/src/21978.htm

[11] 日本自動認識システム協会: JAISA(Japan Automatic Identification Systems Association) バイオメトリクス用語集 http://www.jaisa.or.jp/action/group/bio/About%20BIO/yogo.html

[12] 田中 一郎: US Insight Silicorn valley Research 2005.3
http://www.nttdata.co.jp/event/report/usinsight/us_2005/pdf/usi_vol24.pdf

[13] EUROPE: Second-generation Schengen Information System (SIS II)
http://europa.eu/scadplus/leg/en/lvb/l14544.htm
[14] OECD ヨーロッパにおける人の移動：傾向および将来の見通し
http://www.jil.go.jp/foreign/event_r/event/documents/2004sopemi/2004sopemi_session_2.pdf
[15] The Minutiae Template Interoperability Testing http://www.mtitproject.com/index.html
[16] Max Snijder: BioTesting Europe 2007.5
http://www.biotestingeurope.eu/BIGLisbonRevisedPresentation10May.ppt
[17] Biotesting Europe PRESS RELEASE
http://www.biotestingeurope.eu/BioTestingPressReleaseLaunch.doc
[18] Maria Margarida Castro NevesTesting and Certification of Biometric Components and System in Europe http://www.biometricscatalog.org/DocumentMgr/ViewDocument.aspx?pk=748796a8-38c4-4f51-b7d5-951459efd283
[19] FVC2006 http://bias.csr.unibo.it/fvc2006/databases.asp
[20] James L Wayman National Biometric Test Center Collected Works 1997-2000 2000.08
[21] CIO COLUMN2 US-Visitを構成するアプリケーション
http://www.ciojp.com/contents/?id=00002774;t=0
[22] Department of Homeland Security US-VISIT: Current Ports of Entry
http://www.dhs.gov/xtrvlsec/programs/editorial_0685.shtm
[23] マルコス,ネグロン,赤池法律事務所 US VISIT生体認証バイオメトリクス式監視システム
http://www.mnllp.com/entdepgovjpn.html#US%20VISIT%20Details%202003%20(jpn)
[24] Department of Homeland Security US-VISIT Moves Out of Biometric Exit Pilot Phase 2007.04
http://www.dhs.gov/xnews/releases/pr_1178549052332.shtm
[25] Department of Homeland Security US-VISIT Enrollment Requirements
http://www.dhs.gov/xtrvlsec/programs/editorial_0527.shtm
[26] JETRO 通商弘報 2007.12
http://www3.jetro.go.jp/jetro-file/search-text.do?url=13001776
[27] JETRO 米国輸出手続き 2007.5
http://www3.jetro.go.jp/jetro-file/search-text.do?url=010031400205
[28] TSA Frequently Asked Questions
http://www.tsa.gov/what_we_do/layers/twic/twic_faqs.shtm
[29] Cheryl Gerber Military Information Technology Online 2007.10
http://www.military-information-technology.com/article.cfm?DocID=2209
[30] DBEKS https://secure.biometrics.dod.mil/Home/
[31] Department of Defense to issue up to 13 million Common Access Cards for smart card-enabled PKI http://www.smart.gov/information/cac_case_study_full.pdf
[32] CAC FAQ https://www.cac.mil/Faqs.do
[33] 情報処理推進機構IPA: バイオメトリクス評価に関する調査 2005.03
[34] 情報処理推進機構IPA: IC・ID カードの相互運用可能性の向上に係る基礎調査 2007.01

http://www.ipa.go.jp/security/fy18/reports/ICID/needs_rep.pdf
[35] Mary Mitchell: FIPS 201 Evaluation Program Biometric Product Testing
http://www.biometrics.org/bc2006/presentations/Tues_Sep_19/Session_I/19_Mitchell_nist.pdf
[36]GSA: FIPS 201 Evaluation Program Information Day
http://fips201ep.cio.gov/documents/infodayslides10162007.pdf
[37] NIST IAD FRVT 2006 http://face.nist.gov/frvt/frvt2006/frvt2006.htm
[38] NIST IAD ICE Overview http://iris.nist.gov/ice/
[39] FRVT 2006 and ICE 2006 Large-Scale Results
http://www.frvt.org/FRVT2006/docs/FRVT2006andICE2006LargeScaleReport.pdf
[40] NIST IAD MBGC Overview http://face.nist.gov/mbgc/
[41] IBG Company Overview http://www.biometricgroup.com/IBGOverview.pdf
[42] セキュリティシステム推進協議会 参加企業及び機関
http://www.systemneeds.co.jp/paso/fapi/ronbun-5.htm
[43] asahi.com 外国人の指紋採取，20日からスタート
http://www.asahi.com/national/update/1113/TKY200711130430.html
[44] 船員データ電子化検討委員会 船員データ電子化検討委員会最終とりまとめ
http://www.mlit.go.jp/kisha/kisha06/10/100427/01.pdf
[45]瀬戸洋一：価値あるバイオメトリックシステムを構築・運用するための提言，信学会誌，
pp.1025-1030，Vol.90，No.12，2007年12月

６．４　バイオメトリクスに関する適合性評価対象

６．４．１　開発案件

表6-23は，現在ISO/IEC JTC 1/SC37で開発が進められている案件のうち，適合性評価に関係するものを抽出し，各ワーキンググループ（Working Group）ごとにまとめたものである。

表6-23　適合性評価に関係する開発案件

	開発番号 プロジェクト名	開発 状況	内容	（コ）エディタ
WG2	24709-1　Methods and Procedures	FCD 05年7月	BioAPI適合性試験の手法と手続き	Jason Kim　(韓)
	24709-2　Test Assertions for BSPs	FCD 06年1月 37N1474	BSPに対するアサーション試験	Gregory Zektser (米) 中村（日）
	24709-3　Test　Assertions　for BioAPI Frameworks	WD 08年1月	BioAPIフレームワークのアサーション試験	栗田　（日）
	24709-4　Test Assertions for　Biometric applications	CD/FCD 08年1月	生体認証アプリケーションのアサーション試験	栗田　（日）
WG3	29109　Conformance　Testing Methodology for　Biometric Data　Interchange Records　as defined in　ISO/　IEC19794 Biometric Data　Interchange Format　Standard	NP 07年6月	ISO/IEC19794で定義された生態認証データ互換に関する適合性評価手法 生体認証データ互換フォーマット標準	

	29109-1 Generalized Conformance Testing Methodology	FCD 08年1月	適合性試験手法の一般化	John Campbell (加)
	29109-2 Finger Minutiae Data	CD 08年1月	指紋特徴データ	Greg Cannon (米)
	29109-3 Finger Pattern Special Data	WD 08年1月	指紋パターンスペクトルデータ	Omid Jahromi (米)
	29109-4 Finger Image Data	CD 08年1月	指紋画像データ	Elham Tabassi (米)
	29109-5 Face Image Data	CD 08年1月	顔画像データ	Patrick Grother (米)
	29109-6 Iris Image Data	WD 08年01月	虹彩画像データ	UdoMahlmeister(米) John Duagman (英)
	29109-7 Signature / Sign Time Series Data	WD 08年1月	署名時系列データ	Olaf Henniger (独)
	29109-8 Finger Pattern Skeletal Data	WD 08年1月	指紋 骨格データ	Chiristophe Busch (独) 新崎 (日)
	29109-9 Vascular Image Data	WD 08年1月	静脈血管画像データ	Alex Choi (韓) 浜(日)、緒方(日)
	29109-10 Hard Geometry Silhouette Data	WD 08年1月	掌形シルエットデータ	Samir Tamer (米)
	29109-11 Signature / Sign Processed Dynamic Data	NP 29109-11	署名過程の動的データ	John Dale (英)
	29109-12 Face Identity Data	NP 29109-12	顔識別データ	
	29109-13 Voice Data	NP 29109-13	声紋認証データ	Sautav (米) Bhattacharyya()
WG5	29120 Machine Readable Test Data for Biometric Testing and Reporting	NP 29120	生体認証試験およびレポートのための機械可読試験データ	TBD
	29120 - 1 Test Reports	WD 08年1月	試験レポート	Patrick Grother (米)
	29120 - 2 Biometric System Test Status	WD 08年1月	生体認証システム試験ステータス	Patrick Grother (米)
	29120 - 3 Test Input Data Specifications	WD 08年1月	試験入力データ向仕様	Patrick Grother (米)

セキュアデザイン栗田寛久の意見によれば，「2003年9月のローマでのPlenary会議において，将来SC37で開発する適合性評価の規格とCASCO等の他の組織との関連性等の基本的な方式についてRapporteur Group Mee t ingが開催され、SC37での適合性試験評価規格の骨子についてが決定されている。この報告をもとにBioAPIの規格策定作業とほぼ並行してBioAPIの適合性試験評価の規格作成のNP提案がWDとして開始された。私の記憶では，2006年のロンドン会議までは，適合性試験の規格策定の活動はこれのみだったように思わる。2006年頃よりWG3の規格が完成し、適合性試験の規格策定が活発化されてきたようである。」

SC37としては、発足後すぐに適合性試験の規格開発に関しての議論が行われ将来に対しての提言がされ，適合性評価に関して、発足当時より重要性が認識された。

標準化案件は，プログラムインタフェースやデータ構造の仕様に関するもの，テンプレートデータなど仲介するデータの品質に関するもの，アルゴリズムの性能（精度，処理時間）に関するものがある。それに伴い，コンフォーマンスも，仕様，品質，性能の３つに分けられ開発されると考える。

相互接続性に関しては，仕様と品質が重要である。性能に関しては，アプリケーションの適用において重要なパラメータである。

参考文献

[1] ISO/IEC JTC1/SC37 Business Plan for JTC1/SC37‘biometrics’Period Overed: July 2006-June 2007

[2] ISO/IEC JTC1/SC37　Ｎ１６４８　SC37 Standing Document 1Programme of Work Updated 2006-06-01

６．５　日本における適合性評価体制

６．５．１　適合性評価における標準化の状況

６．2章で述べたように，適合性評価の仕組みは，供給者が適合性評価主体となる第一者評価活動，購入者が主体となる第二者評価活動に代わって，国際標準に準拠した認定機関から認証を受けた第三者機関が主体となる仕組みへと変化してきた。また，国ごとに定められ，法的な強制力を持つ強制規格も国家間での共通化が進んでいる。この背景には，経済活動のグローバル化や，EUによる域内統合の流れの中で，貿易の障害になることを最小限にするというWTO/TBT協定の国際合意の存在が強く反映している[1] [2]。

特に，WTO/TBT協定により一民間機関であったISOが，WTOの国際条約としての強制力から標準規格の世界基準として存在感を強め，適合性評価においては，例えば，ISO/IEC Guide 60:2004Conformity assessment Code of good practice「適合性評価－適正実施基準」のように，ISO/CASCOが発行する規格が，世界基準として重要性を増している。

第三者適合性評価活動の枠組みは，マネジメントシステムや，情報セキュリティシステムの標準であるISO/IEC15408に関する組織体制を見てもわかるように，製品認証，システム認証などさまざまな分野で利用されるようになっている。日本国内の体制を検討する場合，第一者適合性評価，第二者適合性評価，第三者適合性評価のうち，どの評価方法を選択するかは，対象の性質によると考える。ただし，基本的に国際標準のフレームワークに準拠することが必要と考える。例えば，表6-24に示すように指紋特徴点データ(ISO/IEC 19794-2)，指紋画像データ(ISO/IEC 19794-4)，顔画像データ(ISO/IEC 19794-5)，BioAPI適合性試験の手法と手続きISO/IEC 24709-1）などのAPIやデータの論理的物理的構造などの仕様に関するものと，バイオメトリック性能評価＆報告（ISO/IEC19795）に関するものがある[3]。

表6-24　Ｃｏｎｆｏｒｍａｎｃｅ関係の開発案件

分類	プロジェクト名
仕様	ISO/IEC 19785-2 Common Biometric Exchange Formats Framework-Par 2
	FDIS 24709-1 BioAPI Conformance Testing -Part1 Methods and　Procedures
	FCD 24709-2 BioAPI Conformance Testing -Part2 Test Assertions for Biometric Service Provider
	FCD 24709-3 BioAPI Conformance Testing -Part3 Test Assertions for BioAPI Framework
	FCD 24709-4 BioAPI Conformance Testing -Part4 Test Assertions for Biometric Apprications
	29109-1　Conformance Testing Methodology for Biometric Data ・Interchange Records: Part 1 - Generalized Conformance Testing Methodology
	今後はWG4，6における開発案件（システム仕様）がComformance対象になる可能性あり
精度	今後はISO/IEC 19795　Biometric Performance Testing and Reportingなどの開発案件が Comformance対象になる可能性あり
品質	今後はNP29794 Biometric Sample Qualoty Standardsなどの開発案件が Comformance対象になる可能性あり

今後は，バイオメトリクス関係も米国で進むFIPS 201 (Federal Information Processing Standards Publication 201, Personal Identity Verification of Federal Employees and Contractors)に提示されているように、製品，サービスの評価(Evaluation)と認定(Approval)まで標準化のターゲットになると思われる。

適合性評価の方法論に関しては，上記のように方法論の規格開発の議論が中心で，評価体制の整備やMRAのような議論はまだでていない。しかし，米国，EU，韓国で，自国内での体制整備が進んでいる。例えば，米国ではNPIVP(NIST Personal Identity Verification Program)，EUでは，BioTesting Europe Project，韓国ではK-NBTC組織が先進的な取り組みを行っている。

６．５．２　社会的な背景

バイオメトリクスに関する社会的な背景を分析する。生体認証（バイオメトリック認証）の世界市場は，米国の調査会社IBG社（International Biometric Group）によると，2005年は15億ドルを超えた程度だったのが，2006年には20億ドルを超え，2007年は30億ドルを突破して30億1200万ドルに達する見込みである。そして5年後の2012年には74億ドルに達すると予測している。

一方，日本における市場は，日経コンピュータ（2007年11月26日号）によれば，2005年は152億円，2006年には232億円，4年後の2010年には400億円を超えると予測している。これらの数字から日本の市場規模は世界の約10%程度と見積もることができる。日本の市場の特徴は静脈認証装置が市場を席巻している点にある[4]－[6]。

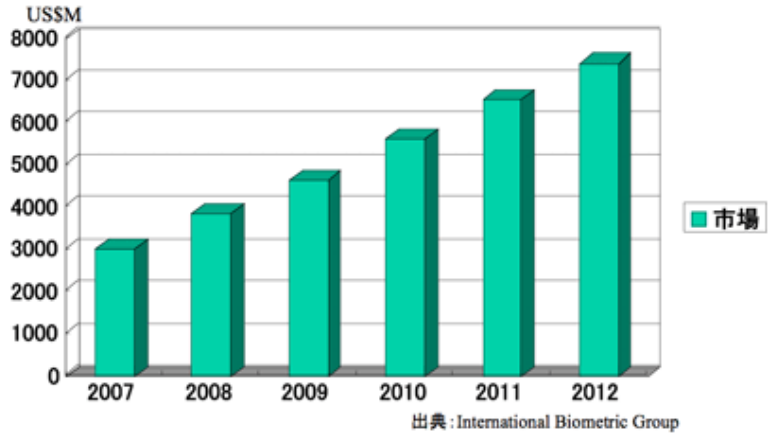


図6-32　生体認証装置の世界市場規模

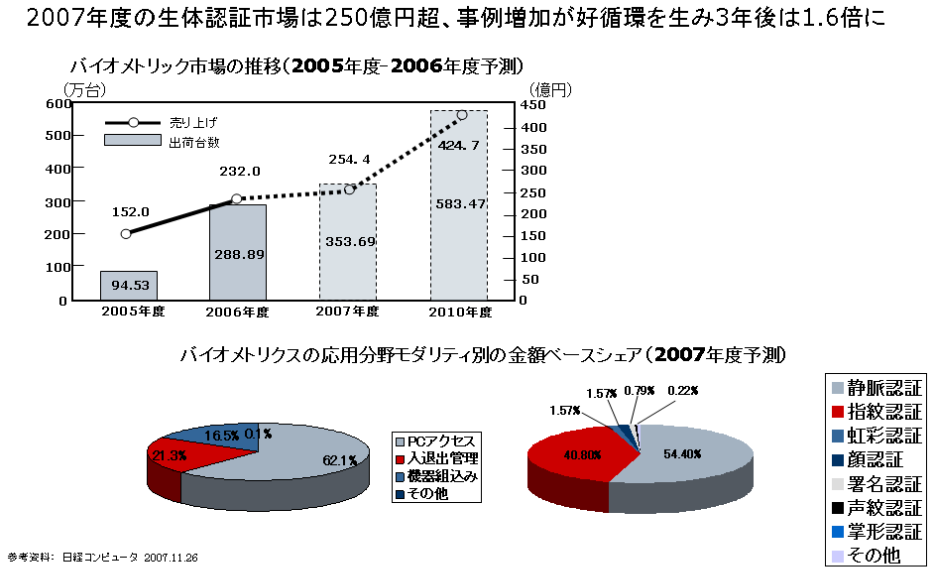


図6-33　生体認証装置の日本市場の規模

市場規模の推移は上記に示したが，生体認証市場がどのような背景で形成し，今後どのような推移をするかが分析することが重要である。図6-34に日本における社会および技術のロードマップを示す[7][8]。

図6-34　生体認証技術のロードマップ

生体認証市場は，3年おきに市場エポックが生じ，市場が拡大してきた。

（1）　第1のエポック（2001年）：　ナショナルセキュリティへの関心増大

米国における同時多発テロの起った2001年9月11日を境に，国境での厳密な人の把握が必要になり，バイオメトリクス技術の重要性が認識された。それ以前は，バイオメトリクス技術は使えるか使えないのかといった議論がされてきたが，この日を境に，なぜ利用しないのかという議論になり，利便性用途からセキュリティ用途に大きくシフトした。

モダリティごとの画像処理技術が進歩した。また，製品値段も10年で100分の１になる一方，出荷台数は100倍以上になるなど市場のパラダイムシフトが生じた（図6-35参照）[１]。　つまり，一部の人たちの技術でなく，社会一般に浸透した技術の流れができてきた。

バイオメトリクス技術が電子パスポート用途への展開などがあり，相互運用などの必要性があり，標準化が重要視され始めた。

（2）第2のエポック（2004年頃）：　金融市場の立ち上がり

日本において，現状のプラスチックベースのキャッシュカードおよび４桁PINの本人確認におけるセキュリティの脆弱性が金融において問題となった。つまりスキミング被害*が大きくなり社会的問題となった。この対策のひとつとしてバイオメトリクス技術に注目があつまり，ATMに静脈認証装置が搭載され本人認証機能が強化された。基幹的民生製品への大規模な初めての展開である。

ちなみに，金融における静脈認証装置は，2004年から3年間の累積売り上げ金額は，約300億円である。ATMへの搭載は約10万台（登録端末含む），バイオメトリクス格納ICキャッシュカード

は，500万枚が発行されている。静脈認証技術の展開は日本独自の傾向である。

画像処理技術だけではなく，センサや装置の一体化などの技術が重要になった，またSC37などの国際標準化委員会で，相互運用や精度評価に関する標準が発行された。

＊スキミング（Skimming）は，近年のカード犯罪で多く使われる手口の一つで，磁気カードに書き込まれている情報を抜き出し，まったく同じ情報を持つカードを複製する犯罪である。

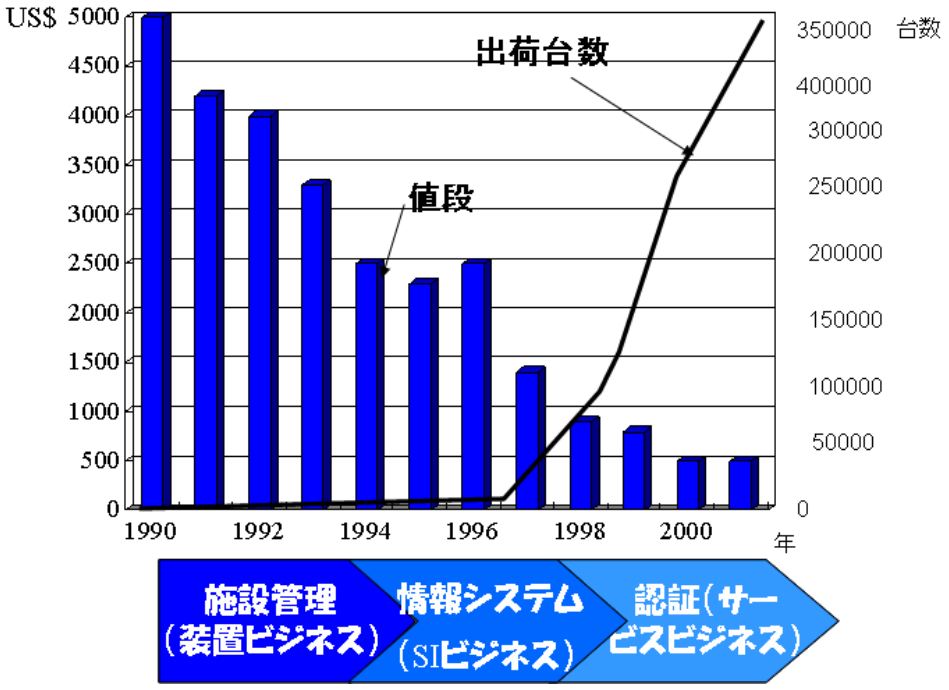


図6-35　製品価格と出荷量の関係

（3）第3のエポック（2007年頃）：　官におけるフォレンジックシステムの普及

国際的な社会IDカード整備のフレームワークに従い，電子パスポート，入国管理システム，ICカード運転免許証など，顔，指紋認証（識別）技術を適用したシステムが，SC37などで開発された標準に準拠し行政主導でシステム構築された。

電子パスポートなどグローバルな実利用が正式に立ち上がったため，適合性の考えが重要になり，プロジェクト単位での適合性評価が行われるようになった（各国の状況に関しては説明済み）。

過去6年の市場の特徴として，エポック2，3となるにつれ，個人利用に浸透してきたことにある。この傾向は今後も強まると考える。つまり，個人端末による本人認証である。現在も携帯電話，モバイルPCへの指紋認証装置が実装され製品化されている。しかし，現在の製品はベストソリューションではなく，問題も多いのが現状である。例えば，一例を示すと，モバイルPCに指紋認証装置がついてセキュリティが強化されているとセールストークになっている製品は，PCが盗難された場合，鍵（PCなどに遺留される指紋パターン）と錠（PCなどに実装された指紋認証装置）

が一緒になっているため，安全とは言えない。特に指紋は遺留性と未対応の問題が大きい。また，パスワードなどのログインモードも併置しているため，単にアクセスルートが複数になっただけで安全性の強化には繋がっていないと考えるのは妥当である。

この類の製品が市場で消費者を混乱させているのが現状である。つまり，現状の製品は安全という観点では問題があり，今後本格的にセキュリティ課題を検討する必要はある。画像処理，信号処理技術もあるレベルにあり，セキュリティを正しく理解した上で，実装面の技術開発が重要である。

したがって，今後5年以内に生ずる，第4のエポックは以下のようになると考える。

（4）第4のエポック（2010年以降）：　生体認証製品のコモディティ化

個人認証として，個々人に生体認証が一般化する。つまり，特別な装置でなく身の回りの製品となる。携帯電話が10年前には特別なものであったのが，現在はなくてはならないＩＴ機器となったのと同様にコモディティ化すると考えられる。

このためには，3つの実装技術開発が重要である。

① 高度なセンサーの開発

② セキュリティの確保された実装開発

③ 生体認証の長所が生かせるアプリケーションプロファイルの開発

コモディティ化により，ユーザに代わり中立的な第三者が安全性などの評価認証を行うことが重要であり，標準の実装，適合性の評価の重要性がでてくる。

エポック2，3における，日本の問題点は，政府関係の電子パスポート，入国管理システム，民間では金融関係システムで，大規模なシステム構築が行われたが，残念ながら，日本には戦略的な政策にかけていたため，適合性評価組織をつくためのノウハウなどが蓄積されていない点にある。本件は，米国，EUと比較して顕著な問題点となっている。

この原因は，構築に際し，オープンな議論をせず，一部のベンダー任せになったことと，取りまとめる公的機関がなかったとこによる。特に，電子パスポートほか政府関係に関しては，いくつかのプロジェクトが，複数の（社団，財団，協会など）法人に分散して調査研究を実施したことが，適合性評価制度をインキュベーションできなかった一因といえる。別の観点から言えば，バイオメトリクスに関しての公的な受け皿となる組織がなかったことが原因といえる。

６．５．３　日本のとるべき対応

適合性評価に関しては，第１適合性から第2，第3者適合性評価がある。どの評価体制をとるかは，評価対象を考慮して検討が必要である。例えば，精度評価などの性能に関しては，評価のためのサンプルがセンサに依存するため，ひとつの評価機関で対処するには，コストがかかる。また，バイオメトリック認証装置は，画像処理技術，判定ポリシーノウハウ，センシングなど，性能には複数の要素が絡むため，開発ベンダーは，画一的な評価機関に性能評価をゆだねることに躊躇する。米国で1990年台サンノゼ州立大学に設置したNBTCは，学術的な成果は得られたが，上記理由により，組織としての運営は成功しなかった。したがって，精度などの性能評価に関す

る適合性評価は，強いユーザによる第2者適合性が適切であると判断する。

米国などで実施されている第2者適合評価に相当する性能評価に関するコンテストは，公開競争による産業育成の側面ももっている。また，ユーザである政府がベンダーのセールストークに惑わされない，製品の実態を把握する発注側主導で発注仕様を作成することができるなどのメリットもある。ただし，日本において上記のような米国と同様の体制で効果的な対応ができるか否かは疑問も残る。

一方，APIなどの仕様に関しては，評価すべきはプログラムインタフェース仕様のため，比較的客観的な評価が可能であり，第3者適合性評価が，発注側および開発ベンダーにも有効である。ただし，評価機関を設置する場合は，その国に市場があるか否かと，輸出製品を提供可能な優れた開発ベンダーが存在するかなどの条件が必要である。したがって，アジアにおいては，日本に適合性評価機関を設置することが，適切であり，日本の製品をアジアの他の国で評価を行うという考えは問題あると考える。

米国などの適合性評価機関の利用も考えられるが，フォレンジックなどの特定市場で立ち上がる諸外国と異なる市場の立ち上がりを行い，また，静脈など日本独自の製品があるため，欧米系の評価機関を利用するということは，産業戦略上考えにくい。

適合性評価に関する提言をまとめると以下のとおりとなる。

(1) 適合性評価は，コモディティ化する製品をユーザに代わり安全性評価などを行う上で重要と考える。ただし，評価機関の設置は，標準化を含めた海外の動向を見極めて進める必要がある。評価組織の拙速な設置は注意必要である。

(2) 海外では，電子パスポート，政府調達関連システムで，各国独自の適合性評価を構築運営に着手し始めている。評価組織とは別途，適合性評価のプロジェクト化によるノウハウなどの獲得は急を要する。検討にあたっては，検討組織を１つにまとめ，設置戦略，ノウハウ，コスト評価の蓄積を行う必要がある。

(3) ノウハウ蓄積のため，日本が主導する精度評価，BioAPIを中心に具体的な活動を行う。一つのきっかけとして2009年に日本で開催予定のABC（Asia Biometric Consortium）カンファレンスのイベントとしてコンフォーマンステストプロジェクトを実施することが有効である。現時点で，トップダウンで進める有効なイベントプロジェクトは，ABCが最適である。これは日本のアジアにおけるリーダシップを発揮する観点でも重要なイベントである。

(4) 産業界，ユーザである行政，企業の意向を聞き最適な評価体制を構築する。案としては、表6-25のようになる。例えば，性能に関しては，図6-36に示す自己評価体制，また，仕様に関しては，図6-37に示す第３者評価が適切である。

表6-25　日本における適合性評価のフレームワーク

対象		評価フレームワーク	備考
相互接続性	BioAPI	第3者適合性評価機関によるテスト認証	
	品質	第3者適合性評価機関によるテスト認証	

	精度	開発ベンダによる精度評価に関し，第3者適合性評価における手順認証	
その他	システム仕様 セキュリティ仕様	・　第2者適合性評価 ・　あるいは、 ・情報セキュリティにおけるCMVPやCCの第3者適合性評価機関によるテスト認証	

理由は，精度などの性能は，センサで取得するデータに依存し，一般的な標準データベースで評価できるのはアルゴリズムなどに限られる。したがって，評価は開発したベンダーで行うがコストの面からも適正であり，第3者評価は，標準にあった精度評価を行っているかなどの手順評価が適切である。手順評価が適正な場合は，評価手続きの適正を認証する仕組みが適している。一方，相互接続に関係するAPIやデータ構造などの仕様は，第3者評価機関で，反証テストなどを行い認証を行う体制案を提案する。

第３者適合性評価機関を構築するか，あるいは，手順評価するか評価機関で反証テストを行うかは別として，テストツールの公的機関での開発は急務である。

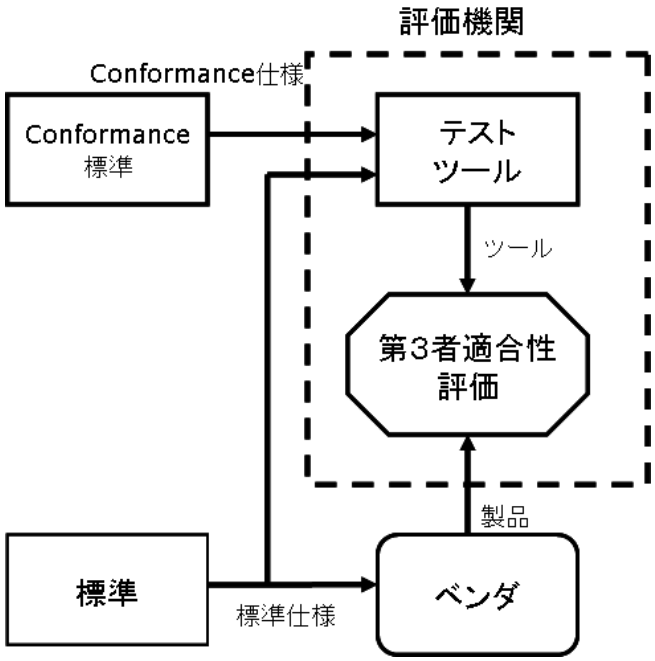


図6-36　自己評価による適合性評価体制

(5) 適合性評価に関しては，現在，API，品質，精度において標準化開発が進むが，ISO/IEC15408などのITシステムのセキュリティ評価との関係や，生体検知機能，バイオメトリクス特有のセキュリティ評価の開発，プラバシー影響評価（Privacy Impact Assessment）などの検討も重要である。

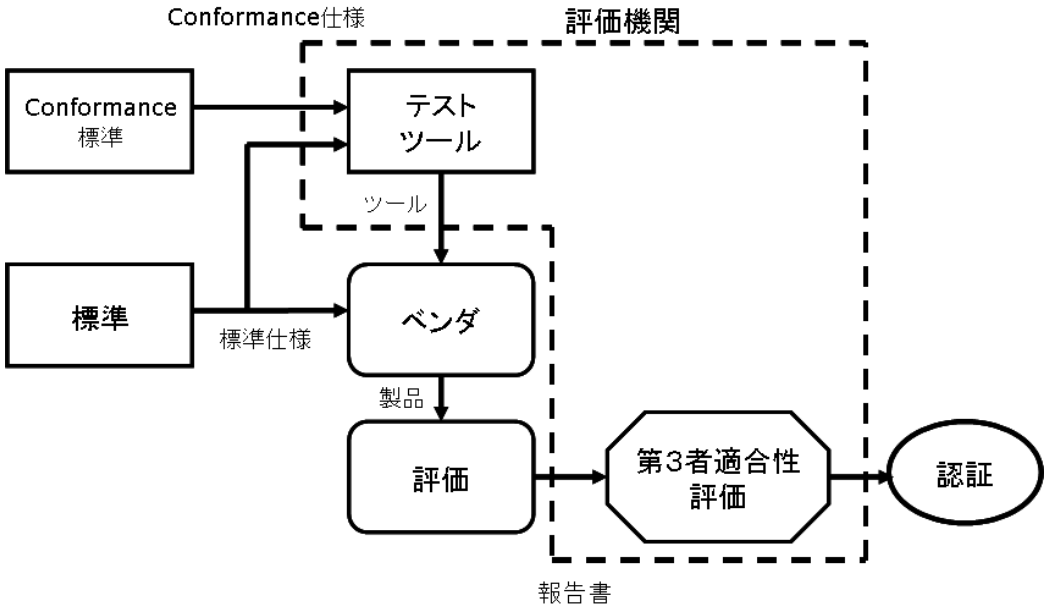


図6-37　第3者適合性評価体制

参考文献

[1]奈良好啓：国際標準化入門、日本規格協会　2004

[2]瀬戸洋一編著： ユビキタス時代のバイオメトリクスセキュリティ 日本工業出版 2003年1月

[3]ISO/IEC JTC1/SC37 N1648 SC37 Standing Document 1 2006-6-1

[4] 瀬戸洋一： バイオメトリックセキュリティ入門 ソフトリサーチセンター2003年

[5] バイオメトリクスセキュリティコンソーシアム： バイオメトリックシステムセキュリティハンドブック

オーム社 2006年

[6] 日経コンピュータ 2007年11月26日号

[7] 瀬戸洋一： 価値あるバイオメトリックシステムを構築・運用するための提言，信学会誌 pp.1025-1030，Vol.90，No.12，2007年12月

[8]瀬戸洋一：バイオメトリック技術の課題と対策，ナノICTシンポジウム2008，2008年2月13日

６．６　まとめ

本調査では，適合性評価制度の実体を国際標準のフレームワークおよび海外の先行事例などを調査することにより明らかに，日本におけるISO/IEC JTC1１/SC37に関する適合性評価体制およ

び体制構築の戦略的なシナリオを，日本の産業力強化の観点から策定した。

バイオメトリック市場は，2001年より3年おきにエポックが生じている。2001年：米国の同時多発テロにより，バイオメトリック技術が市民権を得た。2004年：市民権は得てはいたが停滞した市場において，銀行カードのスキミング不正使用などにより本人認証機能の強化が行われ，静脈技術の実用化により，民生利用における本格的な利用が立ち上がった。2007年：2006年の電子パスポート発行に始まり，IC運転免許証，入国管理システムなどの社会公共システム市場が本格的に立ち上がった。2008年以降は，バイオメトリック技術がコモディティ化し，モバイルPCや端末などにおける本格導入が考えられる。つまり，装置の小型化と標準化がさらに進む。この場合，利用者に代わって，中立的な第3者による評価検証が重要となる。

海外で，適合性評価機関を設置したのは，韓国のみである。韓国ではKorea-National Biometric Centerが設置されている。他の国は明確な組織はないが，米国では，精度評価などが開発ベンダに代わりIBG社がTrusted Third Partyとして評価を実施している。欧州ではSIS2の開発に同期し，Biotestingプロジェクトを実施し，実質的な適合性評価ノウハウを蓄積している。シンガポールでは，ASTARなどの研究機関によりコンフォーマンスコンテストなどが行われ，組織はないが，設置に関する，測定ツールの開発や適合性評価のノウハウは確保されている。

一方日本では，いくつかの社会基盤システムの開発において，実施あるいは設置する機会はあったが，長期的な戦略のもとにプロジェクトが実施されていないため，ツールの開発や組織設置に関するノウハウが蓄積されていない問題がある。

適合性評価に関しては，第１適合性から第2，第3者適合性評価がある。どの評価体制をとるかは，測定する評価対象を考慮して検討が必要である。例えば，精度評価などの性能に関しては，評価のためのサンプルがセンサに依存するため，ひとつの評価機関で対処するには，コストがかかる。また，バイオメトリック認証装置は，画像処理技術，判定ポリシーノウハウ，センシングなど，性能には複数の要素が絡むため，開発ベンダーは，画一的な評価を行う機関に性能評価をゆだねることを躊躇する。米国で1990年台サンノゼ州立大学に設置したNBTCは，学術的な成果は得られたが，上記理由により，組織としての運営は成功しなかった。したがって，精度などの性能評価に関する適合性評価は，可能ならば，強いユーザによる第2者適合性が適切であると判断する。あるいは，評価方法が国際標準に準じて行われたか否か手順認証を行うことが適切であると判断する。

一方，APIなどの仕様に関しては，評価すべきはプログラムインタフェース仕様のため，比較的客観的な評価が可能であり，第3者適合性評価が有効である。ただし，評価機関を設置する場合は，その国に市場があるか否かと，輸出製品を提供可能な優れた開発ベンダーが存在するかが重要な考慮用件となる。

アジアにおいては，静脈など日本独自の製品があるため，他国の評価機関を利用するということは，産業戦略上考えにくい，日本に適合性評価組織を設置することが，適切である。設置には，市場および国際状況を見極めることが重要である。ツール開発やノウハウの把握は重要であるが，

拙速な設置は，かえって市場の拡大にブレーキをかける問題がある。

アジアにおける適合性に関するイニシアチブと，ツールの開発，運用ノウハウを蓄積するため，2009年に日本で開催予定されていうるABCにおいて，適合性評価コンテストを実施することが有効と考える。